

AD-A135 855

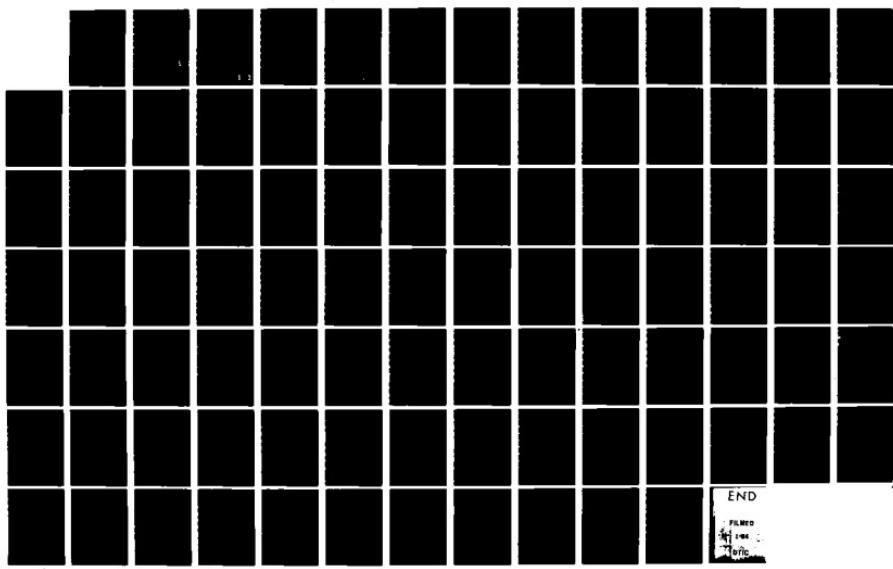
IMPLICIT NUMERICAL SOLUTION FOR A NORMAL SHOCK(U) AIR
FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOOL OF
ENGINEERING C M ALLEN MAY 83 AFIT/GAE/RE/82D-2

1/1

UNCLASSIFIED

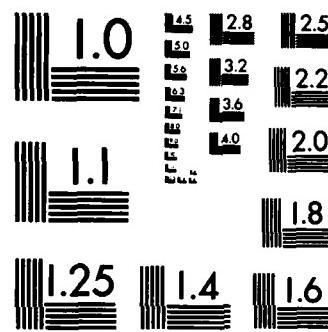
F/G 20/4

NL



END

FILMED
1983
MAY 22
ERIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

AD-A135855



IMPLICIT NUMERICAL SOLUTION
FOR A NORMAL SHOCK

THESIS

AFIT/GAE/AE/82D-2

Charles M. Allen
1st Lt USAF

DISTRIBUTION STATEMENT A

Approved for public release;
Distribution Unlimited

DTIC
ELECTED

DEC 15 1983

S

B

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY (ATC)

AIR FORCE INSTITUTE OF TECHNOLOGY

DTIC FILE COPY

Wright-Patterson Air Force Base, Ohio

83 12 13 243

AFIT/GAE/AE/82D-2

**IMPLICIT NUMERICAL SOLUTION
FOR A NORMAL SHOCK**

THESIS

AFIT/GAE/AE/82D-2

**Charles M. Allen
1st Lt USAF**

Approved for public release; distribution unlimited

**DTIC
ELECTED
S D
DEC 15 1983
B**

AFIT/GAE/AE/82D-2

**IMPLICIT NUMERICAL SOLUTION
FOR A NORMAL SHOCK**

THESIS

**Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University
in Partial Fulfillment of the
Requirements for the Degree of
Master of Science**

**by
Charles M. Allen, B.S.
1st Lt USAF
Graduate Aeronautical Engineering
May 1983**

Approved for public release; distribution unlimited.

PREFACE

Study, development and accomplishment of this thesis has been a great learning experience for me as it introduced me to the full scope of indepth study. The rigors of research: attaining an idea, developing it, and moving forward, (be the result a disappointment and returning to step one, or a success and going on to more indepth concepts), have given me much appreciation and personal committment to the life of a seeker of truth.

I could not give this paper proper preface without acknowledging Capt. James K. Hodge to whose credit I owe the beginning conception of this project. I most sincerely thank Capt. Hodge for his constant interest manifest by his support and insight at times of frustration, and for his caring concern reflected by encouragement, be that in the form of enlightening criticism or compliments.

Individual thanks I also express to my sweet wife, Debbie, for her loving support and assistance. I also thank my children, Joseph and Jamie for their patience with 'Dad' as he reached for his goal of achieving higher learning.



| | |
|--------------------------------------|---|
| Accession For | |
| NTIS | GRA&I <input checked="" type="checkbox"/> |
| DTIC TAB | <input type="checkbox"/> |
| Unannounced <input type="checkbox"/> | |
| Justification _____ | |
| By _____ | |
| Distribution/ _____ | |
| Availability Codes _____ | |
| Distr | Avail and/or Special |
| A-1 | |

TABLE OF CONTENTS

| | Page |
|--|------|
| Preface | ii |
| List of Figures | v |
| List of Symbols | vii |
| Abstract | ix |
| I. Introduction | 1 |
| II. Governing Equations | 4 |
| Normalized Governing Equations | 5 |
| III. Boundary Conditions | 7 |
| Background Information | 7 |
| Characteristic Theory | 9 |
| Nozzle Boundary Conditions | 12 |
| Characteristic Variables | 14 |
| Numerical Conditions | 17 |
| IV. MacCormack's Implicit Algorithm | 20 |
| Derivation | 20 |
| Modification to MacCormack's Implicit Algorithm | 23 |
| Solution Method | 25 |
| Extension to Multi-Dimensions | 27 |
| Boundary Conditions | 30 |
| V. Results | 34 |
| VI. Conclusions | 61 |
| Bibliography | 63 |
| Appendix A: Jacobian Derivation | 65 |
| Appendix B: Characteristic Boundary Conditions--P&U | 67 |
| Appendix C: Characteristic Boundary Condition--P | 70 |
| Appendix D: Sample Calculations for the Initial Shock Location . | 72 |

TABLE OF CONTENTS

| | Page |
|--|------|
| Appendix E: Theoretical Values for Moving the Shock Upstream and Downstream | 75 |
| Appendix F: Program Listing | 77 |
| VITA | 80 |

LIST OF FIGURES

| <u>Figure</u> | | <u>Page</u> |
|---------------|--|-------------|
| 1 | Computational Domain | 10 |
| 2 | Flow of Information | 12 |
| 3 | Flow of Information | 12 |
| 4 | Discrete Computational Mesh | 18 |
| 5 | Differencing Scheme | 19 |
| 6 | Mach Number Distribution for a Supersonic Nozzle: Implicit Scheme | 36 |
| 7 | Pressure Distribution for a Supersonic Nozzle: Implicit Scheme | 37 |
| 8 | Mach Number Distribution for a Supersonic Nozzle: Explicit Scheme | 38 |
| 9 | Pressure Distribution for a Supersonic Nozzle: Explicit Scheme | 39 |
| 10 | Mach Number Distribution for Subsonic Flow with a Shock: Implicit, BC:P | 40 |
| 11 | Pressure Distribution for Subsonic Flow with a Shock: Implicit, BC:P | 41 |
| 12 | Mach Number Distribution for Subsonic Flow with a Shock: Implicit, BC:P&U | 42 |
| 13 | Pressure Distribution for Subsonic Flow with a Shock: Implicit, BC:P&U | 43 |
| 14 | Mach Number Distribution for Subsonic Flow with a Shock: Explicit, BC:P | 44 |
| 15 | Pressure Distribution for Subsonic Flow with a Shock: Explicit, BC:P | 45 |
| 16 | Mach Number Distribution for Subsonic Flow with a Shock: Explicit, BC:P&U | 46 |

| <u>Figure</u> | | <u>Page</u> |
|---------------|--|-------------|
| 17 | Pressure Distribution for Subsonic Flow with a Shock: Explicit, BC:P&U | 47 |
| 18 | Mach Number Distribution for Subsonic Flow with a Shock: Explicit, BC:P&U, Δt = Variable | 48 |
| 19 | Pressure Distribution for Subsonic Flow with a Shock: Explicit, BC:P, Δt = Variable | 49 |
| 20 | Mach Number Distribution for Moving the Shock Upstream: Explicit, BC:P | 52 |
| 21 | Pressure Distribution for Moving the Shock Upstream: Explicit, BC:P | 53 |
| 22 | Mach Number Distribution for Moving the Shock Downstream: Explicit, BC:P | 54 |
| 23 | Pressure Distribution for Moving the Shock Downstream: Explicit, BC:P | 55 |
| 24 | Mach Number Distribution for Trying to Move the Shock Upstream: Explicit, BC:P, Δt = Variable | 56 |
| 25 | Pressure Distribution for Trying to Move the Shock Upstream: Explicit, BC:P, Δt = Variable | 57 |
| 26 | Mach Number Distribution for Trying to Move the Shock Downstream: Explicit, BC:P, Δt = Variable | 59 |
| 27 | Pressure Distribution for Trying to Move the Shock Downstream: Explicit, BC:P, Δt = Variable | 60 |

LIST OF SYMBOLS

| <u>Symbol</u> | <u>Description</u> |
|-----------------|--|
| A | area |
| A(U) | Jacobian |
| c | speed of sound |
| C _E | constant related to the Courant Number |
| D | eigenvalue matrix |
| e | total energy per unit volume |
| F | flux vector |
| H | source vector |
| J | points in computational domain |
| k | coefficient of heat conductivity |
| L | nozzle length |
| P | pressure |
| Q _i | characteristic variables |
| T | temperature |
| t | time |
| U | vector of conservative variables |
| u | velocity in x-direction |
| u ₀ | reference velocity |
| v | velocity in y-direction |
| X | transformation matrix |
| X ⁻¹ | eigenvector matrix |

| <u>Symbol</u> | <u>Description</u> |
|---------------|---------------------|
| x | x-distance |
| y | y-distance |
| ρ | density |
| ρ_0 | reference density |
| λ_i | eigenvalues |
| γ | specific heat |
| μ | viscosity |
| v | kinematic viscosity |

ABSTRACT

There are basically two techniques used to solve the Navier-Stokes equations for fluid flow. These techniques are implicit and explicit methods. Both methods along with characteristic boundary conditions for solving quasi-one-dimensional nozzle flow are presented. Two types of characteristic boundary conditions, those of Steger and McKenna, were tested for each scheme. Solutions of isentropic supersonic flow and flow with shocks were obtained for a diverging nozzle. The behavior of each boundary condition on both implicit and explicit schemes were the same. They deviated from the theoretical values by less than one percent. A test to determine the utility of each scheme was run by allowing the exit boundary conditions to change in the hope of forcing the shock to move upstream or downstream. The shock would not move in the implicit scheme for either boundary condition. The explicit scheme could move the shock, but only when Steger's boundary condition was used. (This is the one which specified only pressure.)

I. INTRODUCTION

During the last decade much progress has been made both in the development of computer systems that are more powerful and reliable and in numerical techniques of solving the compressible Navier-Stokes equations. In spite of this rapid progress in these two related areas, we still cannot calculate the flow fields past complete aircraft configurations (e.g., commercial transport, fighter aircraft, reentry vehicles, etc.) at flight Reynolds numbers. If we could efficiently calculate the flow field, there would be no need to conduct wind tunnel testing or build expensive experimental test vehicles. These costly experimental devices could all be supplanted by simply solving the compressible Navier-Stokes equations. Once the technique of solving these equations is perfected, the result will be a dramatic decrease in the design cost of new aerodynamic vehicles. Since we are not yet able to efficiently solve the compressible Navier-Stokes equations, continued research in this area is important. The process of developing bigger and more powerful computers is one aspect in solving the Navier-Stokes equations, but the development of numerical algorithms that efficiently and accurately solve these equations is the motivation behind this paper.

To date a number of algorithms have been written that accurately solve for some flow fields. These algorithms however, are only accurate for simplified configurations or components of complicated configurations. Three important factors that must be considered in developing

a numerical algorithm are: first, its convergence rate (how fast it can solve a problem); second, its robustness or reliability which means providing correct answers for a variety of problems; third, its ability to adapt to complicated geometries in either two or three dimensions. This criteria was and still is the guide for developing a new and better numerical algorithm.

Basically, algorithms can be classified into two categories: explicit and implicit methods. During the 1950's and 1960's several explicit finite-difference algorithms were developed such as Lax-Wendroff¹ types and the popular MacCormack's method.¹⁰ These methods were very popular and capable of handling a variety of problems. They were able to solve high Reynolds number flow problems as well as inviscid-viscous interactions. During the 1970's however, researchers developed implicit methods to gain better computational efficiency over the explicit methods. Again, both the implicit and explicit methods have enjoyed some degree of success, but each have their own unique limitations. For example, the convergence rate of explicit methods are highly sensitive to space mesh size and in order to improve the solution, the space meshes must be refined in areas such as boundary surfaces, shocks, and stagnation points. Consequently, the computer time needed to converge to a solution becomes very large. Also, explicit methods are conditionally stable. Therefore, in order to calculate high Reynolds number problems a large amount of computing time is required due to the small time step limitation imposed by the stability condition. On the other hand, implicit methods are unconditionally stable and are not restricted to explicit stability conditions. However, they are more complicated than explicit methods and are not as robust. They also have problems in handling the inviscid region as well as with shock

capturing. Today the major emphasis is with implicit methods because they do not require as much computer time as explicit methods. (MacCormack and Lomax² as well as Hollanders and Viviand³ have presented papers that compare the relative pros and cons of the explicit and implicit methods as well as their many variations.)

The purpose of this paper is to study implicit techniques used in solving the compressible Euler equations. Areas such as boundary conditions, differencing techniques and shock capture are addressed. Using the criteria of convergence rate, robustness, and adaptability, the advantages and disadvantages of implicit methods will be determined.

II. GOVERNING EQUATIONS

Throughout this paper the test equations used are the quasi-one-dimensional Euler equations. The one dimensional Euler equations are used because of their simplicity to demonstrate the principles presented. The extension to two and three dimensions as well as the full compressible Navier-Stokes equations is straightforward. It is important to stress the fact that if the extension to three dimension were not straightforward, then that particular method would be useless. Useless, because it cannot be applied to realistic problems.

The quasi-one-dimensional unsteady compressible form of the Navier-Stokes equations, neglecting body forces and heat sources, can be written in conservation form as

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} + \mathbf{H} = 0 \quad (1)$$

where

$$\mathbf{U} = \begin{pmatrix} \rho \\ \rho u \\ e \end{pmatrix} \quad (2)$$

and

$$\mathbf{F} = \begin{pmatrix} \rho u \\ \rho u^2 + \sigma \\ u(\sigma + e) - kT_x \end{pmatrix} \quad (3)$$

and

$$H = \begin{pmatrix} 0 \\ -p \frac{\partial A}{\partial x} \\ 0 \end{pmatrix} \quad (4)$$

and where

$$\sigma = p - \lambda \frac{\partial u}{\partial x} - 2\mu \frac{\partial u}{\partial x} \quad (5)$$

but by using Stoke's hypothesis ($\lambda = -\frac{2}{3}\mu$)

$$\sigma = p - \frac{4}{3}\mu \frac{\partial u}{\partial x} \quad (6)$$

The vector F is the flux vector while U is the vector of conservative variables and H is the source term. The governing equations are in the "weak" conservation form because the source term (H) is outside of the derivatives of the conserved variables. The primitive variables are density ρ , pressure P , and velocity u . The total energy per unit volume is represented by e and defined as

$$e = \frac{P}{\gamma-1} + \frac{\rho u^2}{2} \quad (7)$$

where gamma (γ) is the ratio of specific heats. The viscosity coefficients are λ and μ while the coefficient of heat conductivity and temperature are represented as k and T respectively.

Normalized Governing Equations

The governing equations are normalized so that the characteristic parameters of the problem may be independently varied. The reference

condition used is the freestream condition. The dimensional values are normalized as follows

$$\begin{aligned} u &= \frac{\bar{u}}{u_0} & \rho &= \frac{\bar{\rho}}{\rho_0} \\ x &= \frac{\bar{x}}{L} & t &= \frac{\bar{t}}{L/u_0} \\ p &= \frac{\bar{p}}{\rho_0 u_0^2} & e &= \frac{\bar{e}}{\rho_0 u_0^2} \\ \mu &= \frac{\bar{\mu}}{\mu_0} \end{aligned}$$

The continuity equation will be normalized to demonstrate how the procedure is performed.

Given

$$\frac{\partial \bar{p}}{\partial \bar{t}} + \frac{\partial \bar{p}}{\partial \bar{x}} \bar{u} = 0 \quad (8)$$

multiply the whole expression by $\frac{1}{\rho_0 u_0}$ to get

$$\frac{\frac{\partial \bar{p}}{\partial \bar{t}} \frac{\bar{p}}{\rho_0}}{L/u_0} + \frac{\frac{\partial \bar{p}}{\partial \bar{x}} \bar{u}}{L} = 0 \quad (9)$$

or in nondimensional form

$$\frac{\partial p}{\partial t} + \frac{\partial pu}{\partial x} = 0 \quad (10)$$

The momentum and energy equation are nondimensionalized the same way only they are multiplied by $\frac{1}{\rho_0 u_0^2}$ and $\frac{1}{\rho_0 u_0^3}$ respectively.

III. BOUNDARY CONDITIONS

The study of boundary conditions is important in solving any partial differential equation. Even the extension to finite difference equations does not deter from this fact. The way boundary conditions are chosen can influence the result of any numerical scheme (i.e. convergence, stability, and accuracy). In fact, though one might start out with a stable interior scheme (i.e., scheme for the interior points), it can become unstable and inaccurate if the boundary conditions are improperly treated.

In this section numerical boundary schemes are studied. Specifically, those schemes that can be used with an implicit finite difference scheme. Implicit boundary condition procedures have been studied for several years for the purpose of reaching a time-asymptotic steady state faster as well as for allowing a time step that is unrestricted by the Courant, Friedrich, Lewy (CFL) stability criteria.

Background Information

Consider the simple wave equation

$$\frac{\partial u}{\partial t} + c \frac{\partial u}{\partial x} = 0, \quad 0 \leq x \leq 1, \quad t \geq 0 \quad (11)$$

with the initial condition

$$u(x, 0) = f(x)$$

Kreiss⁴ has shown that according to mathematical laws, in order to be able to solve this type of problem, certain analytic boundary conditions

must be specified. These boundary conditions are dependent upon the sign of C (positive or negative). For example, if C is positive then there must be a boundary specified at $X = 0$, or if C is negative then the boundary condition must be specified at $X = 1$. In mathematical terms

$$u(1,t) = g_1(t) \text{ for } c < 0$$

or

$$u(0,t) = g_2(t) \text{ for } c > 0$$

Once this information is specified the solution of this problem can proceed in a straightforward manner.

Now look at a system of two linear wave equations:

$$\frac{\partial u}{\partial t} + c \frac{\partial u}{\partial x} = 0$$

and

(12)

$$\frac{\partial v}{\partial t} + a \frac{\partial v}{\partial x} = 0$$

The two boundary conditions needed to solve this problem are again determined by the signs of 'c' and 'a'. For $c > 0$ and $a > 0$ the boundary conditions are given by specifying u and v at $x = 0$. Problems arise though when 'a' and 'c' have different signs. To demonstrate this suppose $c > 0$ and $a < 0$. This means that the waves in u travel from left to right while the waves in v travel from right to left. Therefore, u must be specified at $x = 0$ for one boundary condition while v must be specified at $x = 1$ for the second boundary condition. Now if u and v are expressed (at the boundaries) in terms of each other, problems arise. For example, let

$$u(x,0) = 1 \quad v(x,0) = 0 \quad (\text{initial condition})$$

As the right running wave 'u' approaches the right boundary ($x = 1$) it causes 'v' (which is in terms of u) to be non-zero. A disturbance is generated each time a wave reaches the boundary. This type of boundary condition is referred to as a "reflective boundary condition".

The examples presented are simple models. It was easy to identify the variables and where they were to be specified. The extension to three coupled equations (i.e. Navier-Stokes) increases the complexity. The choice of variables and how they are specified is no longer clear. Problems with reflective boundary conditions are another problem area. A method is needed to determine how these complicated boundary conditions should be handled. This method should represent the physics and be mathematically correct.

Characteristic Theory

The mathematical theory of characteristics for hyperbolic systems of equations is an important clue in determining how boundary conditions should be constructed. Many researchers have also used this clue (ref. 5-9). Characteristic theory is an important clue because it can be used to determine the number of boundary conditions needed to solve a problem without over-determining it. Lets look again at a model hyperbolic equation

$$\frac{\partial u}{\partial t} + c \frac{\partial u}{\partial x} = 0 \quad (13)$$

and let

$$u = \begin{pmatrix} r \\ s \end{pmatrix}$$

and

$$c = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

From characteristic theory it is known that the eigenvalues of a hyperbolic system of equations (known as the characteristic direction) are used to determine which direction the information propagates at the boundaries. Knowing which direction the information flows is vital in knowing how many or which boundary conditions are needed. For the model equation the eigenvalues (λ) of the 'C' matrix are determined by

$$\begin{pmatrix} 1-\lambda & 0 \\ 0 & -1-\lambda \end{pmatrix} = (1-\lambda)(-1-\lambda) = 0$$

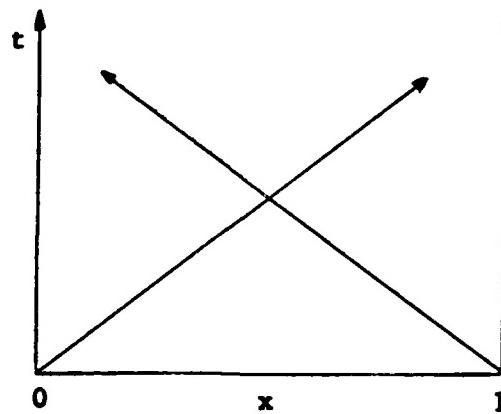
or

$$\lambda^2 - 1 = 0$$

finally

$$\lambda = \pm 1$$

The sign (positive or negative) of the eigenvalues is what determines which direction the information flows (see Figure 1).



Computational Domain

Fig. 1

From this example it is clearly demonstrated (as stated previously) that for positive λ which is associated with r , a boundary condition is needed at $x = 0$ and for negative λ associated with s , a boundary condition is needed at $x = 1$. Another view of characteristic theory is obtained by exploring compatibility conditions. More specifically, a hyperbolic system of equations such as the one-dimensional compressible Navier-Stokes equations is equivalent to three characteristic compatibility conditions. These compatibility conditions are valid along the characteristic curves and are in the form of ordinary differential equations. The slopes of the characteristic curves are given by the eigenvalues. The direction of these characteristic curves, which again are determined by the sign of the eigenvalues, are used in determining boundary conditions. For example, those curves that reach the boundaries from inside the computational domain are considered "admissible". Admissible because they allow information to propagate out of the domain. Those curves that reach the boundary from outside the computational domain are called "inadmissible" because information cannot propagate into the domain. Computational boundary conditions are boundary conditions that are allowed to float or are calculated. They are used with compatibility conditions associated with "admissible" characteristic curves. Specified boundary conditions are boundary conditions that are specified. They are used with those compatibility conditions associated with "inadmissible" curves. In order to understand this concept more easily, look at Figure 2 and Figure 3.

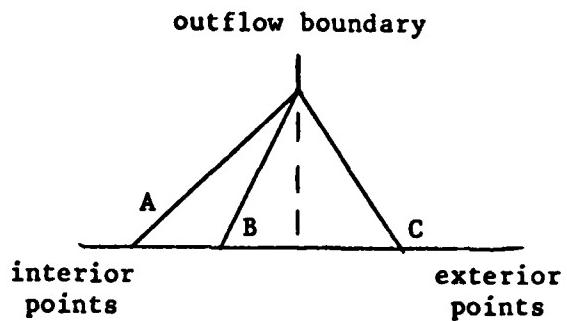
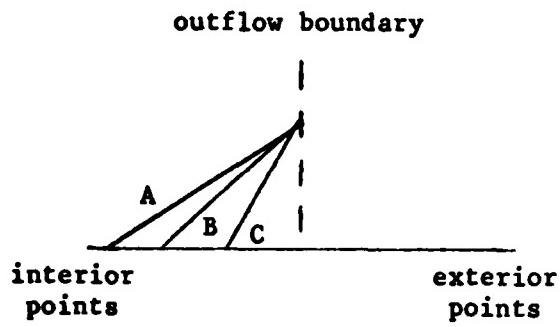


Fig. 2 - Flow of Information

Fig. 3 - Flow of Information

In Figure 2 the characteristic curves of A, B, and C are all admissible, they propagate information out of the computational domain. Therefore the boundary conditions must be calculated for all three compatibility conditions. The sign of the eigenvalues are all positive. In Figure 3 only curve C is inadmissible, therefore a specified boundary condition is needed for this compatibility condition while the boundary conditions for curves A and B are again calculated. Here, λ is positive for curves A and B but negative for the 'C' curve. It is important to note that the eigenvalues do not need to be calculated. The only information needed is the sign (positive or negative). The role of positive and negative eigenvalues at the left boundary are reversed at the right boundary. These concepts can and will be used with the Navier-Stokes equations.

Nozzle Boundary Conditions

The determination of boundary conditions for a nozzle is dictated by characteristic theory. The slope of the characteristic curve or sign of the eigenvalue determines what variables are specified and

which are calculated. As will be shown later, the eigenvalues for the one-dimensional inviscid compressible Navier-Stokes equation are:

$$\lambda_1 = u$$

$$\lambda_2 = u+c$$

$$\lambda_3 = u-c$$

where "u" is the velocity and "c" is the speed of sound. It is evident that the eigenvalues, which determine the boundary conditions, will be determined by the velocity (i.e., subsonic or supersonic).

For nozzle flow, the boundary conditions are simply determined by the velocities at the inflow and at the exit. For the subsonic inflow condition the signs of the eigenvalues are:

$$\lambda_1 = u > 0$$

$$\lambda_2 = u+c > 0$$

$$\lambda_3 = u-c < 0$$

here λ_1 and λ_2 are positive while λ_3 is negative. This corresponds to the slopes of Figure 3 (only this time it is for the inflow boundary). The only characteristic curve that propagates information from the computational domain to the boundary is λ_3 . This means that the variables associated with λ_3 should be calculated while the variables associated with λ_1 and λ_2 should be specified. The variable associated with λ_1 is from the continuity equation. The variables associated with λ_2 and λ_3 are derived from the momentum and energy equations respectively. For the supersonic inflow condition:

$$\lambda_1 = u > 0$$

$$\lambda_2 = u+c > 0$$

$$\lambda_3 = u-c < 0$$

all the eigenvalues are positive which correspond to characteristic curves propagating information to the boundary from outside the computational domain. Therefore all the inflow variables should be specified. Note that the eigenvalues for the subsonic outflow have the same sign as the eigenvalues for the subsonic inflow but the boundary conditions are not the same. They are reversed. For the subsonic outflow according to Figure 3 λ_1 and λ_2 should be calculated while λ_3 should be specified. For supersonic outflow conditions all the variables should be calculated. This was the procedure used in determining boundary conditions for the algorithm.

Note, nothing has been said about the variables associated with the eigenvalues. At first, one would think that the variables should be the primitive variables: density-continuity equation, velocity-momentum equation, and pressure-energy equation. Upon closer examination it is noticed that when the equations (Navier-Stokes) are uncoupled these are not the correct variables.

Characteristic Variables

Characteristic variables are the variables associated with eigenvalues. These are the variables that need to be specified or calculated. The inviscid Navier-Stokes equation representative of one-dimensional flow is given as

$$\frac{\partial U}{\partial t} + \frac{\partial F}{\partial x} + H = 0 \quad (14)$$

Now by using the Jacobian Matrix which is defined as

$$A(U) = \frac{\partial F}{\partial U} \quad (15)$$

The equation can be rewritten as

$$\frac{\partial U}{\partial t} + A(U) \frac{\partial U}{\partial x} + H = 0 \quad (16)$$

The Jacobian matrix can be reduced to

$$A(U) = X D X^{-1} \quad (17)$$

where X^{-1} is the eigenvector matrix and D is the diagonal matrix of the eigenvalues. The Jacobian Matrix ($A(U)$) for these equations (with viscous terms neglected) is given by

$$A(U) = \frac{\partial F}{\partial u} = \begin{bmatrix} 0 & 1 & 0 \\ (\gamma-3) \frac{u^2}{2} & -(\gamma-3)u & \gamma-1 \\ -\gamma u \left(\frac{e}{\rho} \right) + (\gamma-1)u^3 & \left(\frac{e}{\rho} \right) - \left(\frac{3}{2} \right)(\gamma-1)u^2 & \gamma u \end{bmatrix} \quad (18)$$

(see Appendix A for the derivation of the Jacobian)

The eigenvalue matrix is

$$D = \begin{bmatrix} u & 0 & 0 \\ 0 & u+c & 0 \\ 0 & 0 & u-c \end{bmatrix} \quad (19)$$

The transformation matrices are

$$X = \begin{bmatrix} 1 & \alpha & \alpha \\ u & \alpha(u+c) & \alpha(u-c) \\ \frac{u^2}{2} & \alpha \left(\frac{u^2}{2} + uc + \frac{c^2}{\gamma-1} \right) & \alpha \left(\frac{u^2}{2} - uc + \frac{c^2}{\gamma-1} \right) \end{bmatrix} \quad (20)$$

$$\alpha = \frac{\rho}{\sqrt{2}c}$$

and

$$x^{-1} = \begin{bmatrix} 1 - \frac{u^2}{2} \left(\frac{\gamma-1}{c^2}\right) & (\gamma-1) \frac{u}{c^2} & -\left(\frac{\gamma-1}{c^2}\right) \\ \beta((\gamma-1) \frac{u^2}{2} - uc) & \beta(c - (\gamma-1)u) & \beta(\gamma-1) \\ \beta((\gamma-1) \frac{u^2}{2} + uc) & -\beta(c + (\gamma-1)u) & \beta(\gamma-1) \end{bmatrix} \quad (21)$$
$$\beta = \frac{1}{\sqrt{2\rho c}}$$

Now substitute equation 17 into equation 16 to get

$$\frac{\partial u}{\partial t} + \frac{\partial(xDx^{-1})u}{\partial x} = 0 \quad (22)$$

Now multiply the whole equation (22) by x^{-1} to get

$$x^{-1} \frac{\partial u}{\partial t} + x^{-1} \frac{\partial(xDx^{-1})u}{\partial x} = 0 \quad (23)$$

Since x^{-1} is a constant matrix (locally) it can be placed inside the derivative expression.

$$\frac{\partial x^{-1}u}{\partial t} + \frac{\partial Dx^{-1}u}{\partial x} = 0 \quad (24)$$

or

$$\frac{\partial Q}{\partial t} + \frac{\partial DQ}{\partial x} = 0 \quad (25)$$

where

$$x^{-1} u = Q \quad (26)$$

and

$$x^{-1} x = I \quad (27)$$

This equation

$$\frac{\partial Q}{\partial t} + D \frac{\partial Q}{\partial x} = 0 \quad (28)$$

looks like the simple wave equation (12) that was used as a model.

Instead of the variables u and v the variables are now Q_1 , Q_2 , and Q_3 , which is simply the U matrix (vector of conservative variables) multiplied by the eigenvector matrix X^{-1} . The results are

$$X^{-1} U = Q_i = \begin{pmatrix} \rho - \frac{P}{C^2} \\ P + \rho_0 C_0 U \\ P - \rho_0 C_0 U \end{pmatrix} \quad (29)$$

It was assumed that deviations from the free stream are so small that the entries in the Jacobian can be treated as constants-locally. That is where the 0-subscript originates. The eigenvalues associated with 'a' and 'c' are u , $u+c$, and $u-c$.

From the sign of the eigenvalues (u , $u+c$, $u-c$) and the characteristic variables, boundary conditions can be defined.

Numerical Conditions

Numerical boundary conditions are boundary conditions used with finite difference equations. They are the numerical implementation of analytic boundary conditions. Interesting problems arise between the differencing technique and the way boundary conditions are chosen. For example, the way the spatial derivative term $(\frac{\partial u}{\partial x})$ is differenced affects the way boundary conditions are specified. To demonstrate this fact, suppose a central finite difference approximation is used for the spatial term. There are now two points outside the boundaries

that need to be either specified or calculated. Another way to look at it is to look at a discrete mesh (Figure 4)

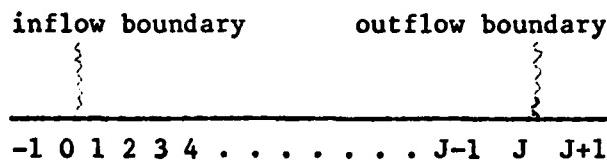


Fig. 4 - Discrete Computational Mesh

with J points in the computational domain. The points -1 (outside the inflow boundary) and $J+1$ (outside the outflow boundary) are outside the computational domain. The value of these points must be made known in order to solve for the J^{th} and 0^{th} point according to the difference relation

$$\left. \frac{\partial u}{\partial x} \right|_J = \frac{(u_{J+1} - u_{J-1})}{2\Delta x} \quad (30)$$

and

$$\left. \frac{\partial u}{\partial x} \right|_0 = \frac{(u_1 - u_{-1})}{2\Delta x} \quad (31)$$

How these values are determined is directly related to the boundary conditions.

As already stated before, the sign of the eigenvalues determines which direction information flows at the boundary. Boundary points are calculated by use of this information. For example, if the eigenvalues are positive at the outflow boundary, a backward differencing scheme should be used. If the eigenvalues are negative then a forward differencing scheme should be used. This is because with positive eigenvalues information flows from inside the computational domain to the boundary. Therefore one would not want to use a forward differencing scheme because then there would not be any continuity

(of information waves) at the boundary. Visually this can be represented by Figure 5.

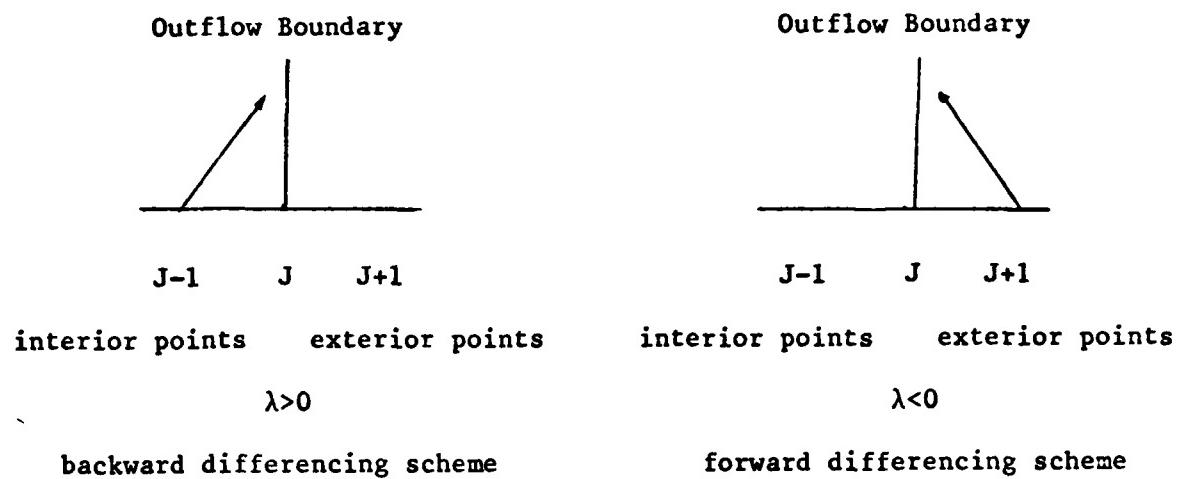


Fig. 5 - Differencing Scheme

IV. MACCORMACK'S IMPLICIT ALGORITHM

MacCormack has developed a new implicit algorithm for solving the inviscid Navier-Stokes equation. It is an extension of his 1969 explicit predictor-corrector algorithm¹⁰. Because it is an extension of an explicit algorithm this new scheme is not truly implicit. For example, it is not unconditionally stable but it does retain a few of the advantages of implicit algorithms such as larger time steps.

The new method¹¹ is composed of two parts. The first part explicitly solves the governing equations by using known flowfield properties to determine local flowfield changes. The second step numerically transforms the equation into an implicit form to remove the explicit stability criteria ($\Delta t = \frac{C N \Delta x}{|u+a|}$). The true flowfield changes are then calculated and substituted into a Taylor series in time to find the new flowfield properties.

Derivation

Differentiating the one-dimensional Navier-Stokes equations with respect to time gives

$$\frac{\partial}{\partial t} \left(\frac{\partial U}{\partial t} \right) + \frac{\partial}{\partial x} \left(\frac{\partial F}{\partial t} \right) + \frac{\partial H}{\partial t} = 0 \quad (32)$$

Using the Jacobian, the above equation can be rewritten as

$$\frac{\partial}{\partial t} \left(\frac{\partial U}{\partial t} \right) + \frac{\partial}{\partial x} \left(\frac{\partial}{\partial t} \left(A \frac{\partial U}{\partial t} \right) \right) + \frac{\partial H}{\partial U} \frac{\partial U}{\partial t} = 0 \quad (33)$$

Implicitly approximating this equation (33) leads to

$$\left(I + \Delta t \frac{\partial A^*}{\partial x} + \Delta t \frac{\partial H}{\partial U} \right) \frac{\partial U^{N+1}}{\partial t} = \frac{\partial U^N}{\partial t} \quad (34)$$

The dot means the derivative operates on the term outside the parenthesis. The equation (34) can now be rewritten as

$$\left(I + \Delta t \frac{\partial A^*}{\partial x} + \Delta t \frac{\partial H}{\partial U} \right) \delta U_i^{N+1} = \Delta U_i^N \quad (35)$$

if the following are defined

$$\delta U^{N+1} = \frac{\partial U^{N+1}}{\partial t} \Delta t \quad (36)$$

and

$$\Delta U^N = \frac{\partial U^N}{\partial t} \Delta t \quad (37)$$

A closer look at equation (35) reveals the two step procedure. The right hand term represents the explicit solution of the governing equations while the left hand side of the equation represents the numerics of the implicit algorithm. Because MacCormack uses a predictor-corrector approach equation (35) is rewritten as predictor:

$$\left(I + \frac{\Delta t}{\Delta x} |A|_i^N + \Delta t \left(\frac{\partial H}{\partial U} \right)_i^N \right) \delta U_i^{\overline{N+1}} = \Delta U_i^N + \frac{\Delta t}{\Delta x} |A|_{i+1}^N \delta U_{i+1}^{\overline{N+1}}$$

where

$$\Delta U_i^N = -\Delta t \left(\frac{F_{i+1}^N - F_i^N}{\Delta x} \right) - \Delta t H_i^N$$

corrector:

$$\left(I + \frac{\Delta t}{\Delta x} |A|_i^{\overline{N+1}} + \Delta t \left(\frac{\partial H}{\partial U} \right)_i^{\overline{N+1}} \right) \delta U_i^{N+1} = \Delta U_i^{\overline{N+1}} + \frac{\Delta t}{\Delta x} |A|_{i-1}^{\overline{N+1}} \delta U_{i-1}^{N+1}$$

where

$$\Delta U_i^{\overline{N+1}} = -\Delta t \left(\frac{F_i^{\overline{N+1}} - F_{i-1}^{\overline{N+1}}}{\Delta x} \right) - \Delta t H_i^{\overline{N+1}}$$

and where

$$\frac{\partial H}{\partial U} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ \frac{(\gamma-1)}{2A} u^2 & \frac{\partial A}{\partial x} - \frac{(\gamma-1)}{A} u & \frac{\partial A}{\partial x} & \frac{(\gamma-1)}{A} \frac{\partial A}{\partial x} \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Spatial derivative terms are approximated by a forward difference for predictor and backward difference for corrector.

The Jacobian (A) is the same as previously defined in equation 17 and 18. The absolute value of the Jacobian ($|A|$) which is used in the predictor and corrector equations is defined as

$$|A| = x \lambda_A x^{-1} \quad (38)$$

(The transformation matrices have already been given, see equations 20 and 21).

The eigenvalue matrix is written as

$$\lambda_A = \begin{bmatrix} \lambda_{A_1} & 0 & 0 \\ 0 & \lambda_{A_2} & 0 \\ 0 & 0 & \lambda_{A_3} \end{bmatrix} \quad (39)$$

where

$$\lambda_{A_1} = \max \left(|u| - CE \left(\frac{\Delta x}{\Delta t} \right), 0.0 \right)$$

$$\lambda_{A_2} = \max \left(|u+c| - CE \left(\frac{\Delta x}{\Delta t} \right), 0.0 \right)$$

$$\lambda_{A_3} = \max \left(|u-c| - CE \left(\frac{\Delta x}{\Delta t} \right), 0.0 \right)$$

CE = a constant related to the Courant number used for the explicit stability.

where

$$\Delta t \leq \frac{CE (\Delta x)}{(u+c)}$$

If the explicit stability criteria is used the λ_A matrix goes to zero. This allows the true flowfield changes to be calculated in the explicit part of the program thereby skipping altogether the implicit part.

Throughout this paper only the inviscid form of the equations were used. If viscous effects are included, the eigenvalue matrix would become:

$$\lambda_1 = |u| + \frac{2v}{\rho \Delta x} - CE \frac{\Delta x}{\Delta t}$$

$$\lambda_2 = |u+c| + \frac{2v}{\rho \Delta x} - CE \frac{\Delta x}{\Delta t}$$

$$\lambda_3 = |u-c| - \frac{2v}{\rho \Delta x} - CE \frac{\Delta x}{\Delta t}$$

(Note: v is nondimensionalized by dividing it by v_0 .)

Modification to MacCormack's Implicit Algorithm

Careful analysis of MacCormack's predictor - corrector equations reveals that after the explicit (local) flowfield changes have been calculated and substituted into the right hand side of equation (35) an upper block bidiagonal system of equations will result for the

predictor step. A lower block bidiagonal system of equations will result for the corrector step. In order for the solution to progress a matrix inversion must be performed at each computational point which will result in a large amount of computer time, thus defeating one advantage of implicit methods.

MacCormack however, has devised an ingenious way to circumvent the inversion process. The $|A|$ matrix, which needs to be inverted, is diagonalized thus making its inversion trivial. Problems with this technique arise when the source term "H" is included in the implicitly approximated governing equations (equation 35). The diagonalization cannot be performed and a matrix inversion must be computed for each point. However, a modification to MacCormack's scheme that will allow a diagonalization by neglecting the source term in the implicit numerics was presented in a paper by White and Anderson¹². They reasoned that the source term "H" could be removed from the implicit numerics because "The physics of the flow is carried by the explicit step (which contains the source term in the governing equations), and the implicit step (which under the present modification does not contain the source term) is simply numerics to bring about enhanced stability." Using their modification the implicitly approximated equation can be rewritten as predictor:

$$\Delta U_i^N = - \Delta t \left(\frac{F_{i+1}^N - F_i^N}{\Delta x} \right) - \Delta t H_i^N$$

$$\left(I + \frac{\Delta t}{\Delta x} |A|_i^N \right) \delta U_i^{N+1} = \Delta U_i^N + \frac{\Delta t}{\Delta x} |A|_{i+1}^N S U_{i+1}^{N+1}$$

$$U_i^{N+1} = U_i^N + U_i^{N+1}$$

corrector:

$$\Delta U_i^{N+1} = -\Delta t \left(\frac{F_i^{N+1} - F_{i-1}^{N+1}}{\Delta x} \right) - \Delta t H_i^{\overline{N+1}}$$

$$\left(I + \frac{\Delta t}{\Delta x} |A| \overline{i}^{\overline{N+1}} \right) \delta U_i^{N+1} = \Delta U_i^{\overline{N+1}} + \frac{\Delta t}{\Delta x} |A| \overline{i-1}^{\overline{N+1}} \delta U_{i-1}^{\overline{N+1}}$$

$$U_i^{N+1} = U_i^N + \frac{1}{2} (\delta U_i^{\overline{N+1}} + \delta U_i^{\overline{N+1}})$$

This is in a form that results in an upper and lower block bidiagonal system which will allow a diagonalization to occur.

Solution Method

As already stated before, this scheme is a predictor - corrector type. The predictor equations are solved first and their results are used in solving the corrector equations. The results from the corrector step, the true flowfield properties, are then used in calculating the predictor step. This process is repeated until the scheme converges. In order to understand this scheme more fully the predictor step will be reviewed in detail. The principles used in the corrector step are the same as those used in the predictor step. The extension therefore should be straightforward.

Initially all the conservative variables "U" are specified (initial guess) for all $i = 1, 2, 3, \dots, J$ (J = is the total number of computational points). Once the end boundary condition is found

$|A|_{i+1}^N \delta U_{i+1}^{\overline{N+1}}$ the right hand side of the predictor equation can be solved and represented as "W"

$$W = \Delta U_i^N + \frac{\Delta t}{\Delta x} |A|_{i+1}^N \delta U_{i+1}^{\overline{N+1}} \quad (41)$$

In order to solve for δU_i^{N+1} , a matrix inversion of $(I + \frac{\Delta t}{\Delta x} |A|_i^N)$ must be performed. By using the relation $A = XDX^{-1}$ the matrix may be diagonalized thereby foregoing the costly (in computer time) matrix inversion. Making this substitution, the equations can be written as

$$\left(I + \frac{\Delta t}{\Delta x} D_A X^{-1} \right) \delta U_i^{N+1} = W \quad (42)$$

Now pre-multiply both sides by X^{-1} to get

$$\left(X^{-1} + \frac{\Delta t}{\Delta x} D_A X^{-1} \right) \delta U_i^{N+1} = X^{-1} W = V \quad (43)$$

Regrouping, the left hand side is now written as

$$\left(I + \frac{\Delta t}{\Delta x} D_A \right) X^{-1} \delta U_i^{N+1} = V \quad (44)$$

The matrix $\left(I + \frac{\Delta t}{\Delta x} D_A \right)$ is diagonal so that its inversion is trivial.

The equations can now be written as

$$X^{-1} \delta U_i^{N+1} = \left(I + \frac{\Delta t}{\Delta x} D_A \right)^{-1} V = Y \quad (45)$$

Where δU_i^{N+1} is found by multiplying the equation by X . The flowfield changes for the point i have just been calculated. Now, in order to calculate the values for the next point ($i-1$) the term $|A|_i^N \delta U_i^{N+1}$ must be calculated. Calculating this term will give all the values for the right hand side of the equation. Once all these values are obtained the procedure used to solve for δU_{i-1}^{N+1} (on the left hand side) is exactly the same as stated before. An efficient way to calculate the $|A|_i^N \delta U_i^{N+1}$ term is to multiply the already known value of Y by D_A then by X . In mathematical terms

$$|A|_i^N \delta U_i^{N+1} = X D_A Y \quad (46)$$

Using this method the predictor step sweeps down from $i = J$ to the $i = 1$ computational point. The corrector step starts at $i = 1$ and marches to the $i = J$ point. Local values are used in calculating the matrices X , D , and X^{-1} at each point.

The solution process can be summarized in the following seven steps:

$$1. \quad W = \Delta U_i^N + \frac{\Delta t}{\Delta x} |A|_{i+1}^N \delta U_{i+1}^{N+1}$$

$$2. \quad V = X^{-1} W$$

$$3. \quad D_A \text{ is calculated}$$

$$4. \quad Y = \left(I + \frac{\Delta t}{\Delta x} D_A \right)^{-1} V$$

$$5. \quad \delta U_i^{N+1} = X Y$$

$$6. \quad Z = D_A Y$$

$$7. \quad |A|_i^N \delta U_i^{N+1} = X Z$$

Because the matrix is diagonal in step 4 the inversion is trivial. Step 5 calculates the flowfield changes while step 7 calculates the flux to be used for the next computational point. If Neumann type boundary conditions are used the $|A|_2^N \delta U_2^{N+1}$ term is saved and used as the boundary condition for the corrector step. The $|A|_{J-1}^{N+1} \delta U_{J-1}^{N+1}$ term is saved from the corrector step and used as the boundary condition for the predictor step. Because of this the boundary conditions are not truly implicit but lag by 1/2 time-step.

Extension to Multi-Dimensions

The value of any one-dimensional scheme is proportional to its ability to adapt to multi-dimensions. A one-dimensional scheme that is extremely accurate but not able to convert to multi-dimensions is

severely limited. The method in devising any numerical scheme is to start out with a simple model, then add upon that model so that it can handle a wider variety of problems. If the model cannot expand then it is a poor model. If the model can expand while still retaining some degree of accuracy then one has a good model. The idea is to start out simple and then become more sophisticated.

MacCormack's implicit scheme is a robust scheme. It can handle a wide variety of problems as well as being able to expand to two and three dimensions. The extension from the simple one-dimensional case to two and three dimensions is straightforward. To demonstrate this fact the scheme will be developed for the two-dimensional case. The general equation form is given by

$$\frac{\partial U}{\partial t} + \frac{\partial F}{\partial x} + \frac{\partial G}{\partial y} = 0 \quad (48)$$

Differentiating the general equation (48) with respect to time and using the Jacobian yields

$$\frac{\partial(\frac{\partial U}{\partial t})}{\partial t} + \frac{\partial A(\frac{\partial U}{\partial t})}{\partial x} + \frac{\partial B(\frac{\partial U}{\partial t})}{\partial y} = 0 \quad (49)$$

(A and B are the Jacobians). Now implicitly approximate the above equation (49) in time to get

$$(I + \Delta t \frac{\partial A \cdot}{\partial x} + \Delta t \frac{\partial B \cdot}{\partial y}) \frac{\partial U^{N+1}}{\partial t} = \frac{\partial U^N}{\partial t} \quad (50)$$

The dots have the same meaning as before (the derivative acts on the term outside the parenthesis). Using the following substitutions

$$\Delta U^N = \Delta t \frac{\partial U^N}{\partial t} \quad (51)$$

and

$$\delta U^{N+1} = \Delta t \frac{\partial U^{N+1}}{\partial t} \quad (52)$$

and by using forward and backward differencing the above equation (50) can be written in predictor-corrector notation as

predictor:

$$\begin{aligned} \Delta U_{i,j}^N &= -\Delta t \left(\left(\frac{F_{i+1,j}^N - F_{i,j}^N}{\Delta x} \right) + \left(\frac{G_{i,j+1}^N - G_{i,j}^N}{\Delta y} \right) \right) \\ \left(I - \frac{\Delta t}{\Delta x} (A_{i+1,j}^N - A_{i,j}) \right) \left(I - \frac{\Delta t}{\Delta y} (B_{i,j+1}^N - B_{i,j}) \right) \delta U_{i,j}^{N+1} &= \Delta U_{i,j}^N \\ U_{i,j}^{N+1} &= U_{i,j}^N + \delta U_{i,j}^{N+1} \end{aligned}$$

corrector:

$$\begin{aligned} \Delta U_{i,j}^{N+1} &= \Delta t \left(\left(\frac{F_{i,j}^{N+1} - F_{i-1,j}^{N+1}}{\Delta x} \right) + \left(\frac{G_{i,j}^{N+1} - G_{i,j-1}^{N+1}}{\Delta y} \right) \right) \\ \left(I + \frac{\Delta t}{\Delta x} (A_{i,j} - A_{i-1,j}) \right) \left(I + \frac{\Delta t}{\Delta y} (B_{i,j} - B_{i,j-1}) \right) \delta U_{i,j}^{N+1} &= \Delta U_{i,j}^{N+1} \\ U_{i,j}^{N+1} &= \frac{1}{2} (U_{i,j}^N + U_{i,j}^{N+1} + \delta U_{i,j}^{N+1}) \end{aligned}$$

The Jacobians are defined (same as before) as $A = X D_A X^{-1}$ and $B = Y D_B Y^{-1}$ where the matrices X , D_A , and X^{-1} are the same as before except for the terms from the Y - momentum equation. The matrices Y^{-1} and D_B are found the same way as X^{-1} and D_A were. They are given as

$$Y^{-1} = \begin{bmatrix} 1 - (\gamma - 1) \frac{(u^2 + v^2)}{2} & \frac{u(\gamma - 1)}{c^2} & \frac{v(\gamma - 1)}{c^2} & -\frac{(\gamma - 1)}{c^2} \\ -\frac{u}{p} & \frac{1}{p} & 0 & 0 \\ \frac{(u^2 + v^2)}{2} (\gamma - 1) - cv & -u(\gamma - 1) & c - v(\gamma - 1) & (\gamma - 1) \\ \frac{(u^2 + v^2)}{2} (\gamma - 1) + cv & -u(\gamma - 1) & -c - v(\gamma - 1) & (\gamma - 1) \end{bmatrix} \quad (53)$$

and

$$D_B = \begin{bmatrix} v & 0 & 0 & 0 \\ 0 & v & 0 & 0 \\ 0 & 0 & v + c & 0 \\ 0 & 0 & 0 & v - c \end{bmatrix} \quad (54)$$

The solution procedure for this scheme in two-dimensions is explained in detail in reference 11. It will not be discussed further.

From this example it has been shown that MacCormack's implicit scheme can be expanded to multi-dimensions. The extension to three-dimensions is straightforward. This little example shows the utility of this scheme.

Boundary Conditions

The boundary conditions used in MacCormack's algorithm are based on characteristic theory. They are applied to a diverging nozzle that has a constant supersonic inflow and an outflow that varies between subsonic and supersonic. Two boundary conditions, one by McKenna¹³ and the other by Steger¹⁴, were studied for their use in subsonic (exit) flows.

The upstream boundary conditions for the supersonic (inflow) nozzle were set by holding the characteristic variables constant. This is because the eigenvalues which are all positive dictate specified boundary conditions. The characteristic variables which are functions of the primitive variables (density, velocity, and pressure) are the

variables that need to be specified. They are specified by holding the primitive variables constant at their inflow values. They are represented as

inflow

| mathematically | numerically |
|--|-----------------------------|
| $Q_1 = \rho - \frac{P}{C_0^2} = \text{constant}$ | $\rho(1) = \text{constant}$ |
| $Q_2 = P + \rho_0 C_0 U = \text{constant}$ | $U(1) = \text{constant}$ |
| $Q_3 = P - \rho_0 C_0 U = \text{constant}$ | $P(1) = \text{constant}$ |

Numerically these boundary conditions were written so that the inflow variables were the same for each time-step.

The downstream boundary conditions for the supersonic (exit) nozzle were calculated. The eigenvalues are positive, thereby indicating right running waves, the same as at the inlet. The difference is that the right running waves approach the boundary from inside the computational domain. The characteristic variables must therefore be calculated. Neumann type boundary conditions were used for this condition. They are represented as

outflow

| mathematically | numerically |
|---------------------------------------|-------------------------|
| $\frac{\partial Q_1}{\partial x} = 0$ | $(Q_1)_j = (Q_1)_{j-1}$ |
| $\frac{\partial Q_2}{\partial x} = 0$ | $(Q_2)_j = (Q_2)_{j-1}$ |
| $\frac{\partial Q_3}{\partial x} = 0$ | $(Q_3)_j = (Q_3)_{j-1}$ |

Using this type of boundary condition insures that waves are not reflected at the boundary¹³.

The exit boundary conditions (Neumann) were implemented into the program by setting the next to last computational point in the corrector step equal to the last point in the predictor step.

$$(Q_i)_{j_{\max}} = (Q_i)_{j_{\max} - 1}$$

Predictor Corrector

This is because the predictor step sweeps from the exit of the nozzle to the entrance while the corrector sweeps from the entrance to the exit. Therefore, the exit boundary condition (used in the predictor step) should come from the corrector step.

The boundary conditions of McKenna and Steger were used for the subsonic (exit) boundary condition. The subsonic boundary condition is more complicated because two of the eigenvalues are positive while the third is negative. This corresponds to having the first two variables calculated while the third is specified. Mathematically it is represented as

$$\frac{\partial Q_1}{\partial x} = 0$$

$$\frac{\partial Q_2}{\partial x} = 0$$

$$Q_3 = \text{Constant} = K3$$

Numerically, in order to specify K3 which is equal to

$$K3 = P - \rho_0 C_0 U$$

the pressure and velocity must be known at the outflow boundary.
 (Remember ρ_0 and C_0 are values used from the previous time step.)
 This is not helpful for real nozzle flow because the only known variable downstream is the pressure. Specifying the velocity downstream is mathematically correct but not realistic. McKenna has used this information in writing his boundary conditions. They are given as

$$\begin{aligned}\rho_j^n &= \rho_{j-1}^n + \frac{\rho_0}{2C_0} \left(\frac{K_3}{\rho_0 C_0 A_j} - \frac{P_{j-1}^n}{\rho_0 C_0} + U_{j-1}^n \right) \\ U_j^n &= \frac{1}{2} \left(U_{j-1}^n + \frac{P_{j-1}^n}{\rho_0 C_0} - \frac{K_3}{A_j \rho_0 C_0} \right) \\ P_j^n &= \frac{\rho_0 C_0}{2} \left(\frac{P_{j-1}^n}{\rho_0 C_0} + U_{j-1}^n + \frac{K_3}{A_j \rho_0 C_0} \right)\end{aligned}$$

where

$$K_3 = P_j - \rho_0 C_0 U_j$$

(The derivation of this boundary condition is in the Appendix)

One assumption that McKenna makes that is suspect is his assumption that K_3 is constant.

A set of subsonic downstream boundary conditions which only specify the pressure were also used. They are Steger's boundary conditions. The derivations of these boundary conditions are also given in the appendix. They are represented as

$$\rho_j = \rho_{j-1} + \frac{1}{C_0^2} (P_\infty - P_{j-1})$$

$$U_j = U_{j-1} + \frac{1}{\rho_0 C_0} (P_{j-1} - P_\infty)$$

$$P_j = P_\infty$$

This type of boundary condition is more common in an experimental setup but allows reflection of waves.

V. RESULTS

A series of computer runs were made using both MacCormack's implicit and explicit algorithms. Boundary conditions were tested for various nozzle conditions to determine their impact on both schemes. A problem of special interest that was investigated was to determine how MacCormack's algorithm (with the help of characteristic boundary conditions) could handle the relationship between the shock location and back pressure.

The results are plotted against the theoretical solution for comparison. Notice that the theoretical shock is smeared. This is because the theoretical solution is calculated and plotted at each computational point. This will give a better comparison between the numerical results and the theoretical solution.

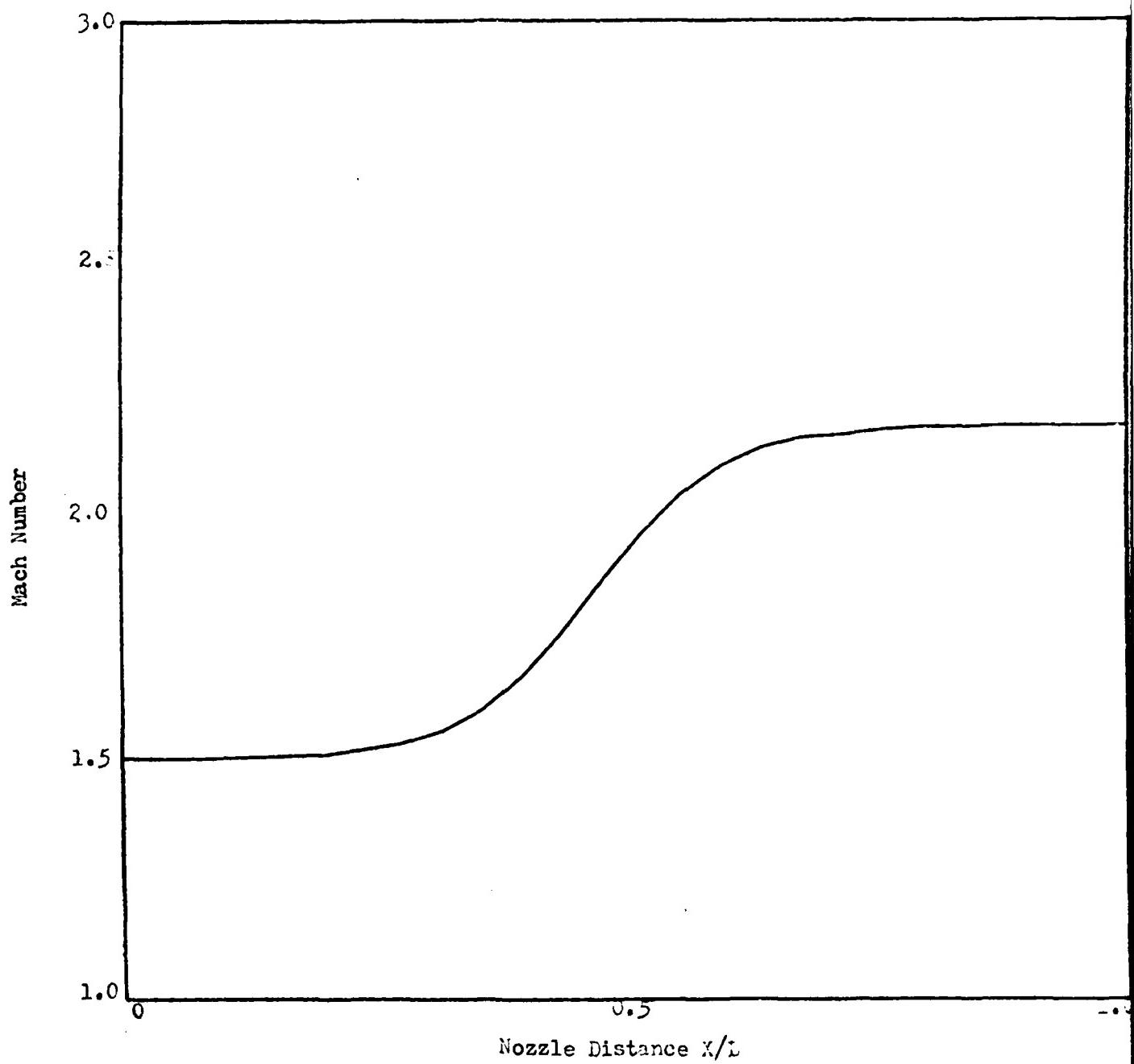
The first test performed was comparing the implicit to the explicit algorithm for the supersonic nozzle. Boundary conditions were not a factor in this computer run because both algorithms handled the conditions the same--numerically. The supersonic downstream boundary conditions were Neumann type. The results were not surprising. The implicit run converged in 135 time-steps while the explicit converged in 1408 time-steps. Interestingly, the implicit run did not converge to the same values of the explicit run. This was a result of a constant which is related to the CFL condition located in the eigenvalue matrix. After this constant was adjusted to 0.9 both algorithms converged to

the same values. These values were within 1% of the theoretical values. The results are plotted on Figures 6 through 9.

The second test performed was a comparison of McKenna's and Steger's boundary conditions. McKenna's was the characteristic boundary condition where both pressure and velocity are specified while Steger's characteristic boundary condition only specified the pressure. Each boundary condition was used in the implicit and explicit algorithm. The subsonic (exit) nozzle was used because characteristic theory states that with subsonic flow there will be one left running wave and two right running waves. This is a perfect condition in which to test the two boundary conditions for their reflection effects on the algorithm. The convergence of both boundary conditions in the implicit case compare quite well. The boundary condition that specified pressure and velocity (McKenna) required one less time-step to converge than Steger's boundary condition. Except for the jumps (Gibb's phenomena) around the shock, both conditions converged to the theoretical values with a deviation of less than one percent. The results are plotted in Figures 10 through 13. The plots for the explicit case are located in Figures 14 through 17. Notice that the Figures are not as smooth as those of the implicit case. After 2,500 iterations the scheme still had not converged. It simply oscillated around the theoretical values. It was suggested that by allowing the time-step Δt to float it would cause the waves to decrease. Accordingly, Δt was set to

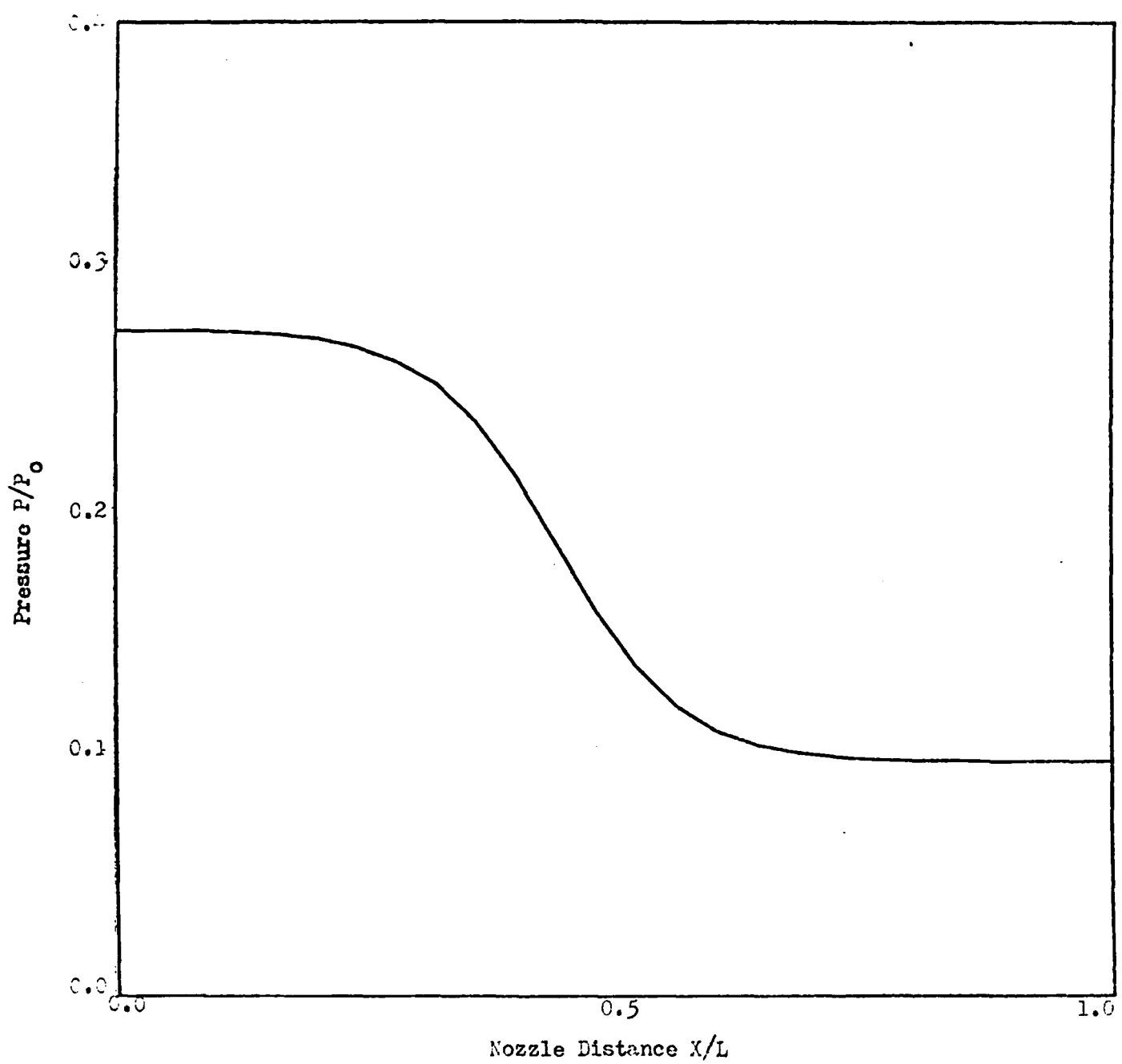
$$\Delta t = \frac{CN \cdot \Delta X}{U + C}$$

for each computational point. The results justified this line of reasoning. Note how smooth the results are--just like in the implicit case (see Figures 18 and 19). The reason the jumps disappeared is



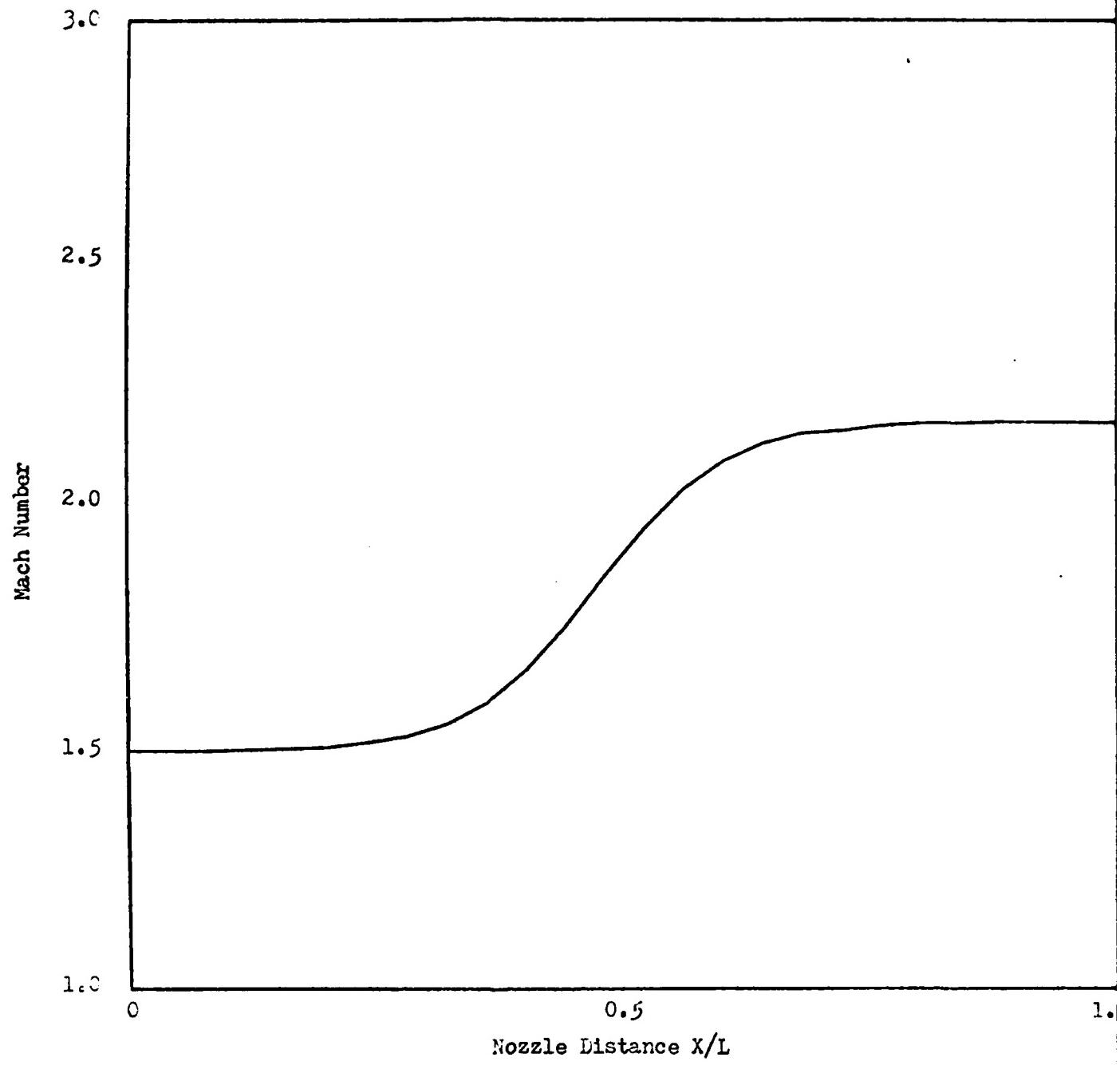
Scheme: Implicit
Nozzle Area: $1.398 + 0.347 \operatorname{Tanh} 0.8(x-4)$
B.C.: Neumann
Theory: - - -
Data Results: —

Figure 6. Mach Number Distribution for a Supersonic Nozzle



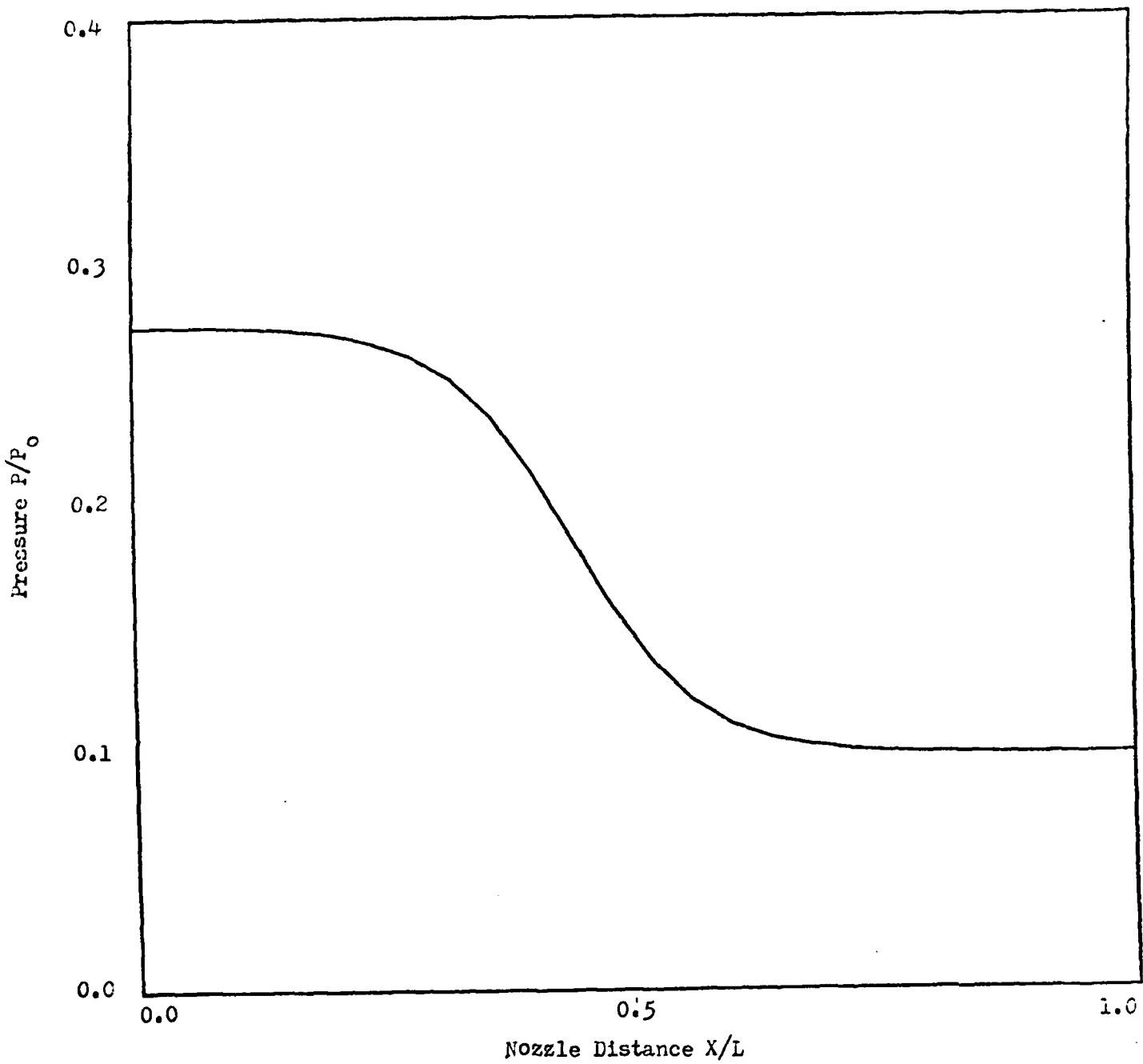
Scheme: Implicit
Nozzle Area: $1.398 + 0.347 \operatorname{Tanh} 0.8(x-4)$
B.C.: Neumann
Theory: — — —
Data Results: — — —

Figure 7. Pressure Distribution for a Supersonic Nozzle



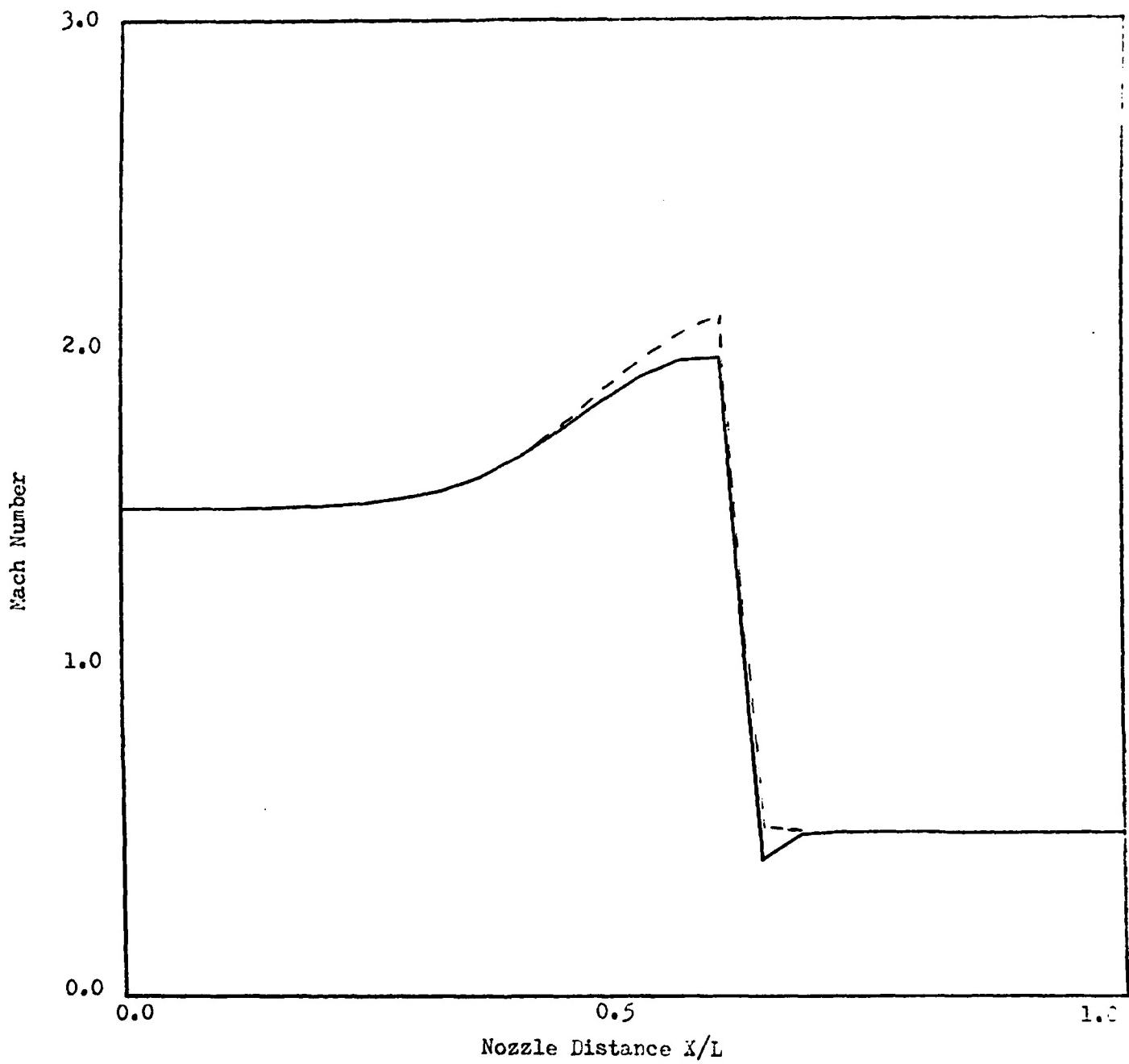
Schemes: Explicit
 Nozzle Area: $1.398 + 0.347 \operatorname{Tanh} 0.8(x-4)$
 B.C.: Neumann
 Theory: ---
 Data Results: ———

Figure 8. Mach Number Distribution for a Supersonic Nozzle



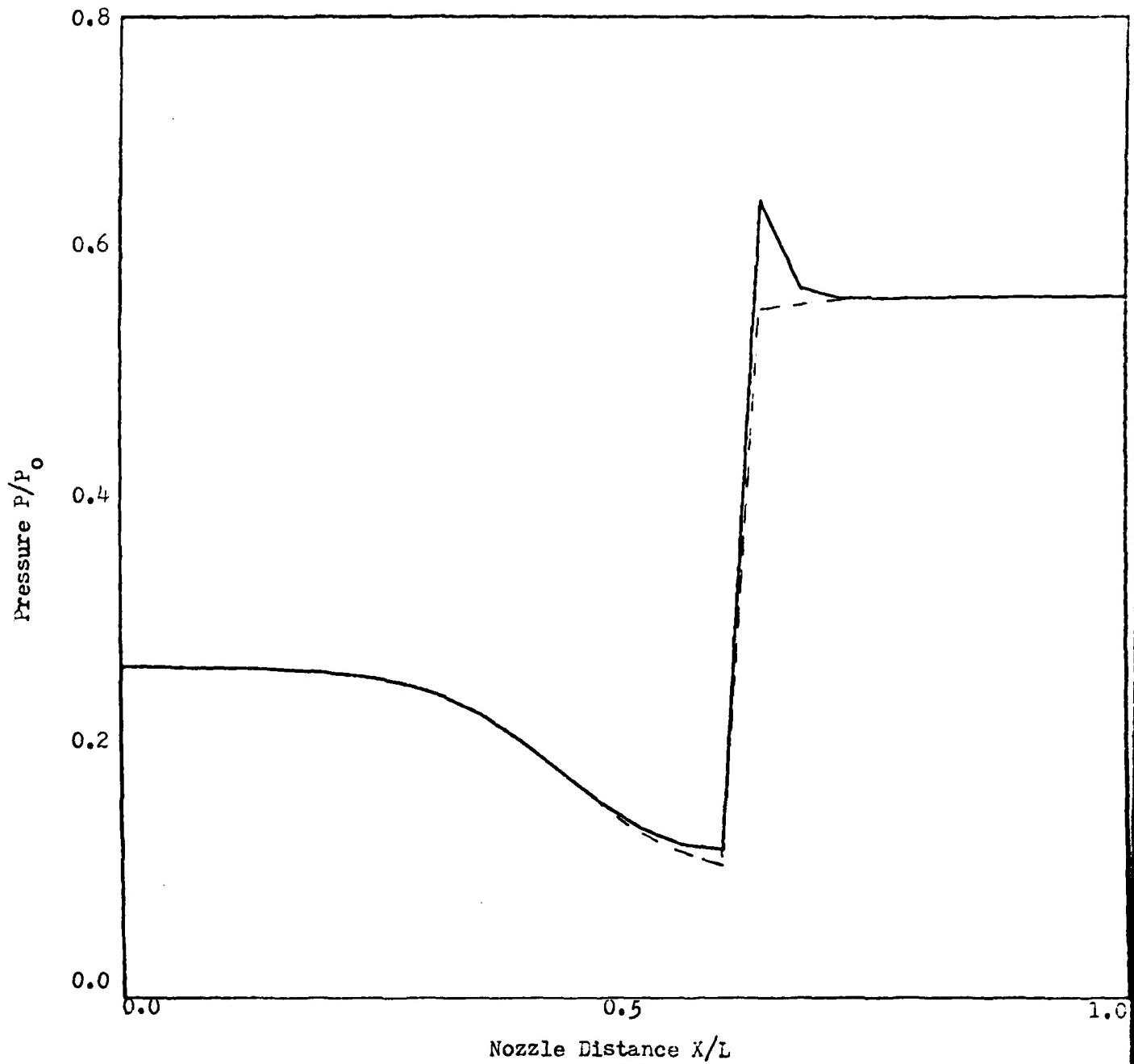
Scheme: Explicit
 Nozzle Area: $1.398 + 0.347 \operatorname{Tanh} 0.8(x-4)$
 B.C.: Neumann
 Theory: — — —
 Data Results: — — —

Figure 9. Pressure Distribution for a Supersonic Nozzle



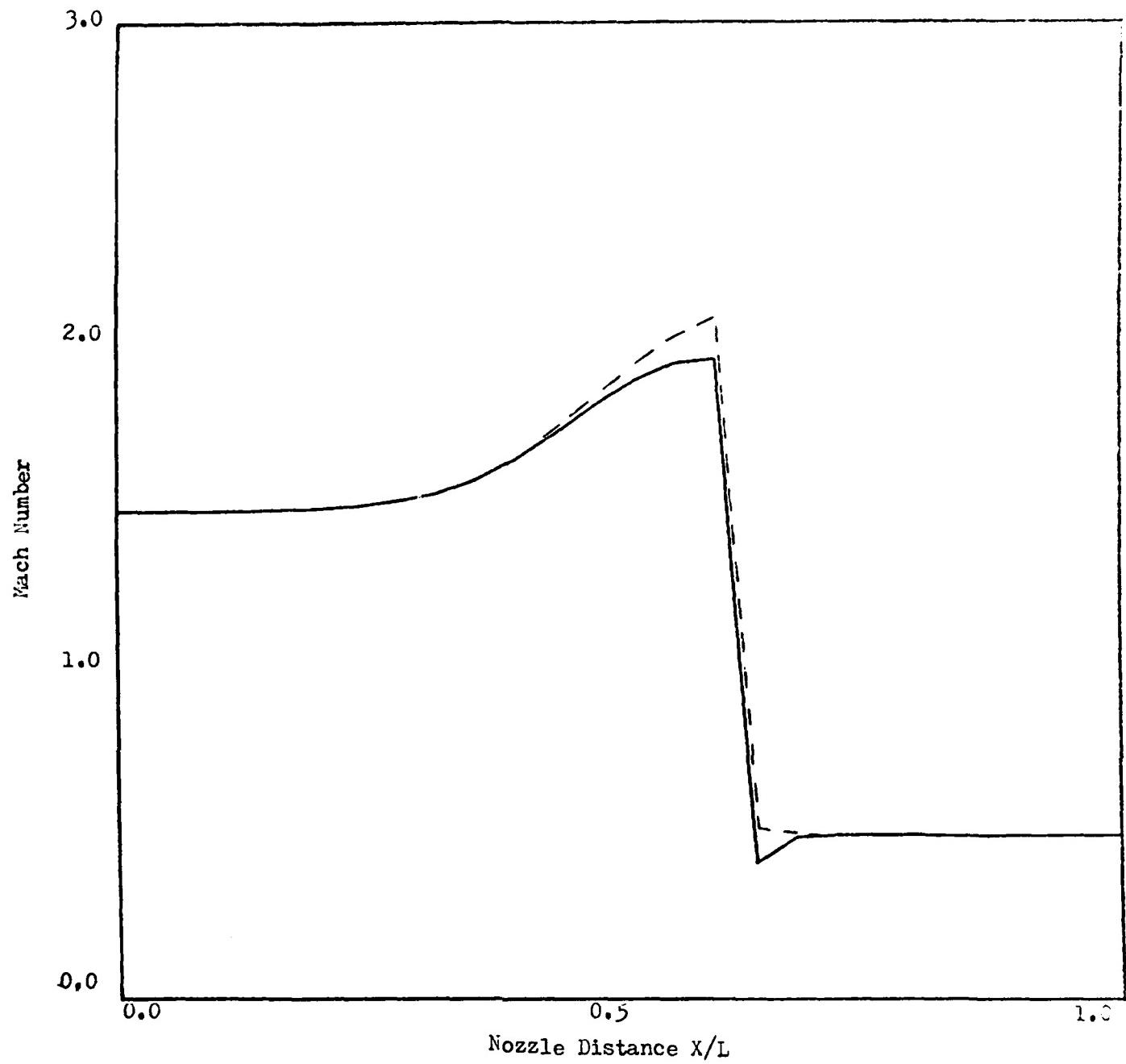
Scheme: Implicit
 Nozzle Area: $1.398 + 0.347 \tanh 0.8(x-4)$
 B.C.: Specify P
 Theory: ——
 Data Results: ——

Figure 10. Mach Number Distribution for Subsonic Flow with a Shock



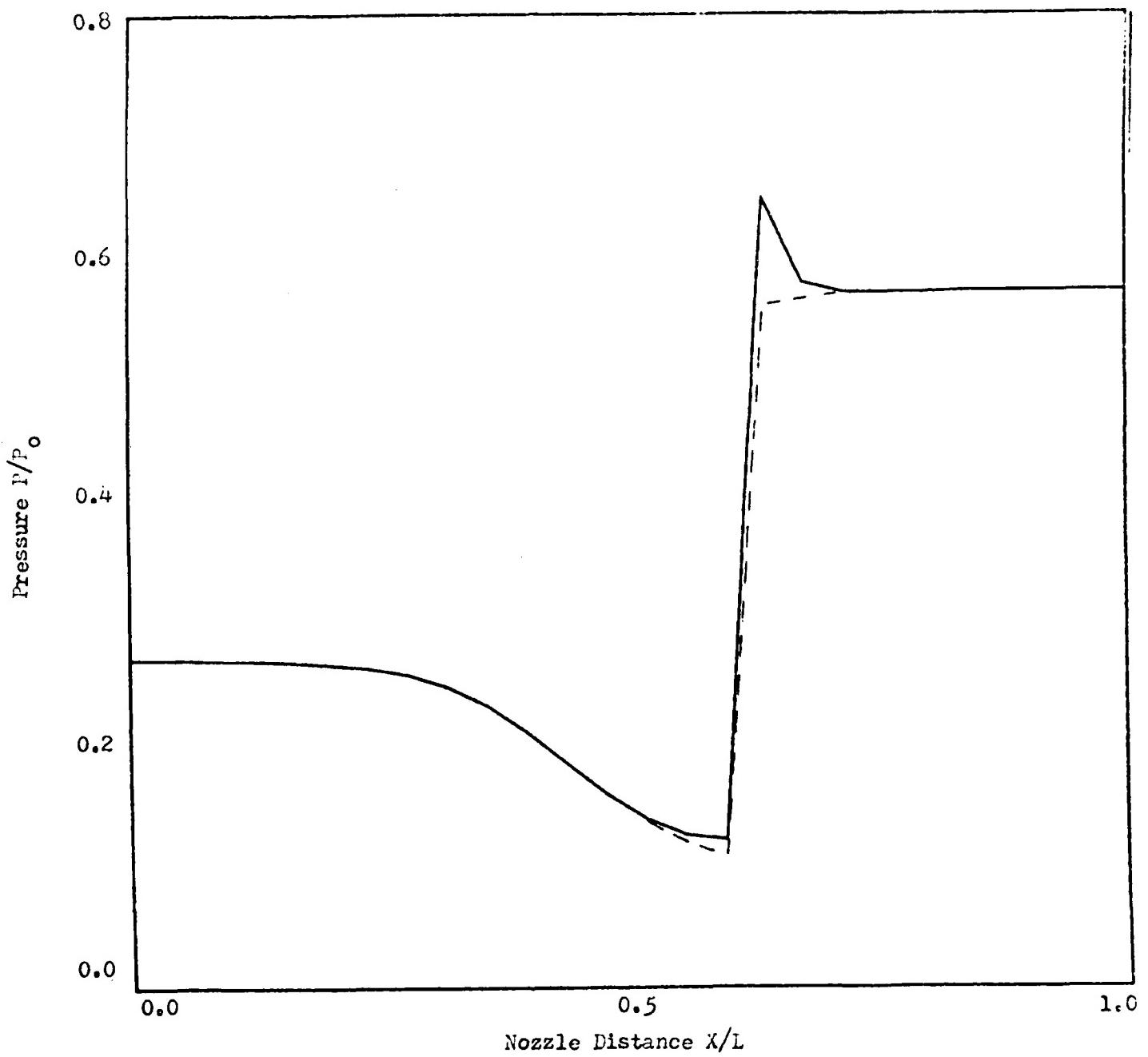
Scheme: Implicit
 Nozzle Area: $1.398 + 0.347 \operatorname{Tanh} 0.8(x-4)$
 B.C.: Specify P
 Theory: — — —
 Data Results: — — —

Figure 11. Pressure Distribution for Subsonic Flow with a Shock



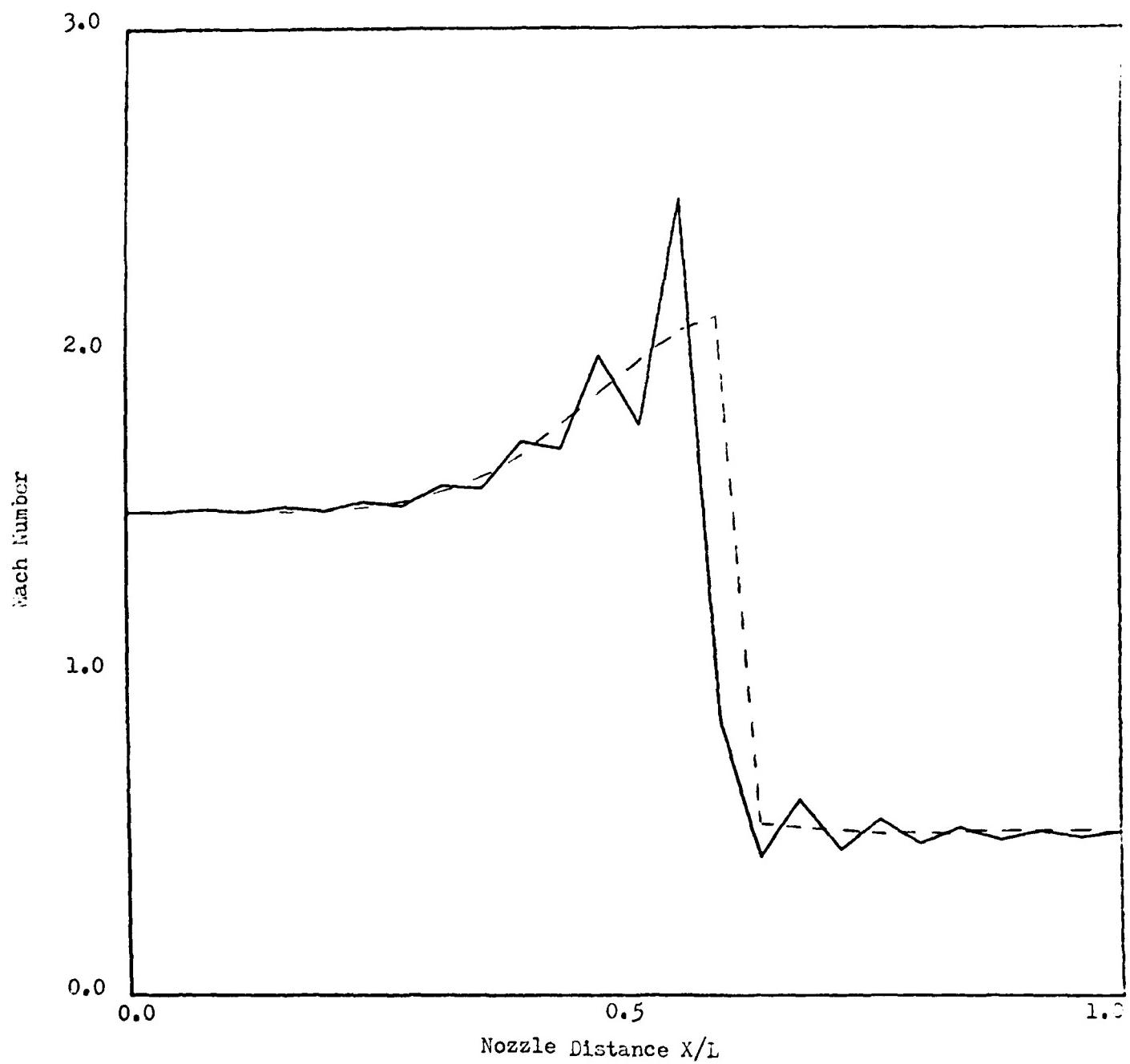
Scheme: Implicit
 Nozzle Area: $1.398 + 0.347 \operatorname{Tanh} 0.8(x-4)$
 B.C.: Specify P_{∞}
 Theory: — — —
 Data Results: — — —

Figure 12. Mach Number Distribution for Subsonic Flow with a Shock



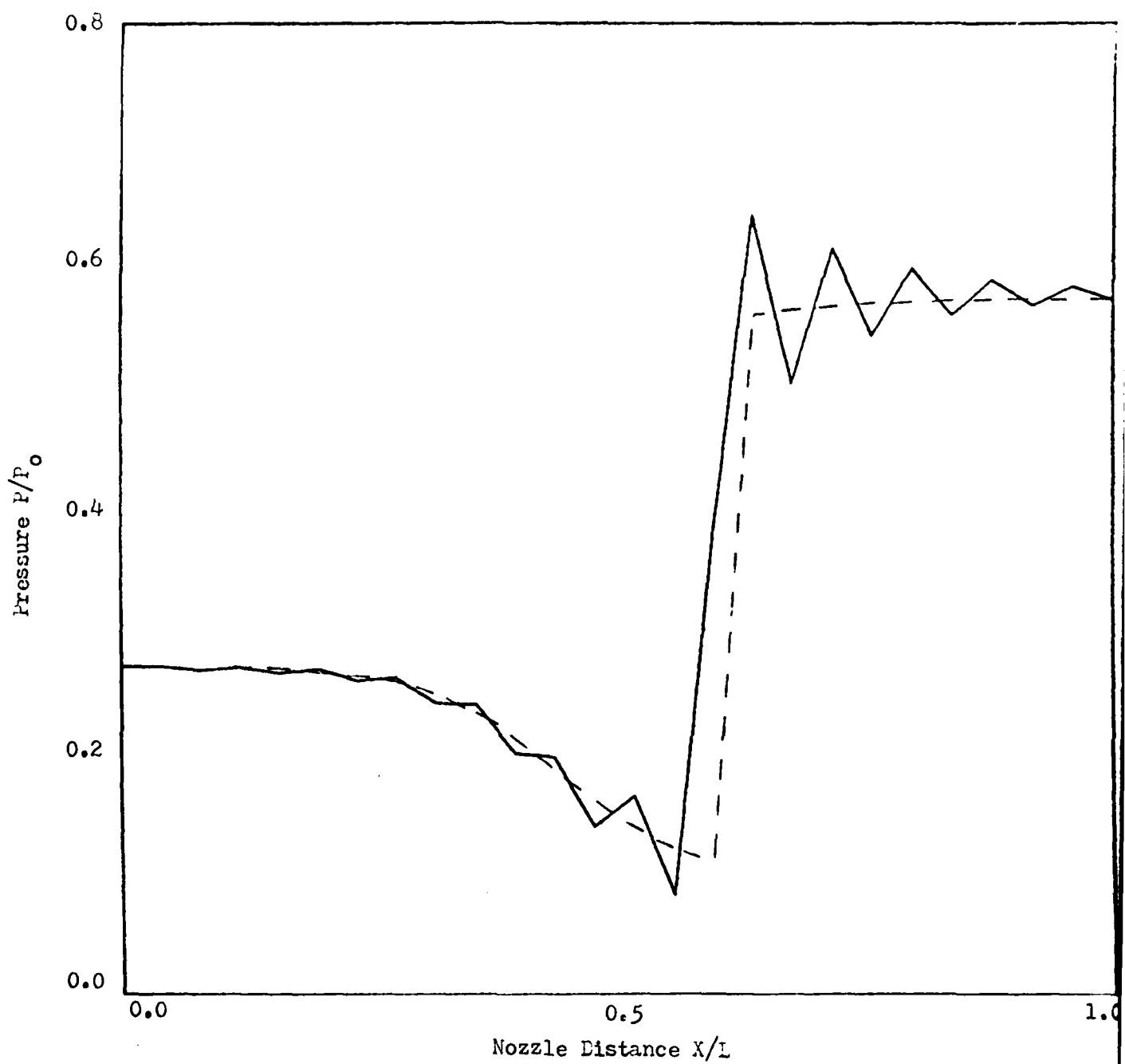
Scheme: Implicit
 Nozzle Area: $1.398 + 0.347 \operatorname{Tanh} 0.8(x-4)$
 B.C.: Specify $P\&U$
 Theory: ——
 Data Results: ———

Figure 13. Pressure Distribution for Subsonic Flow with a Shock



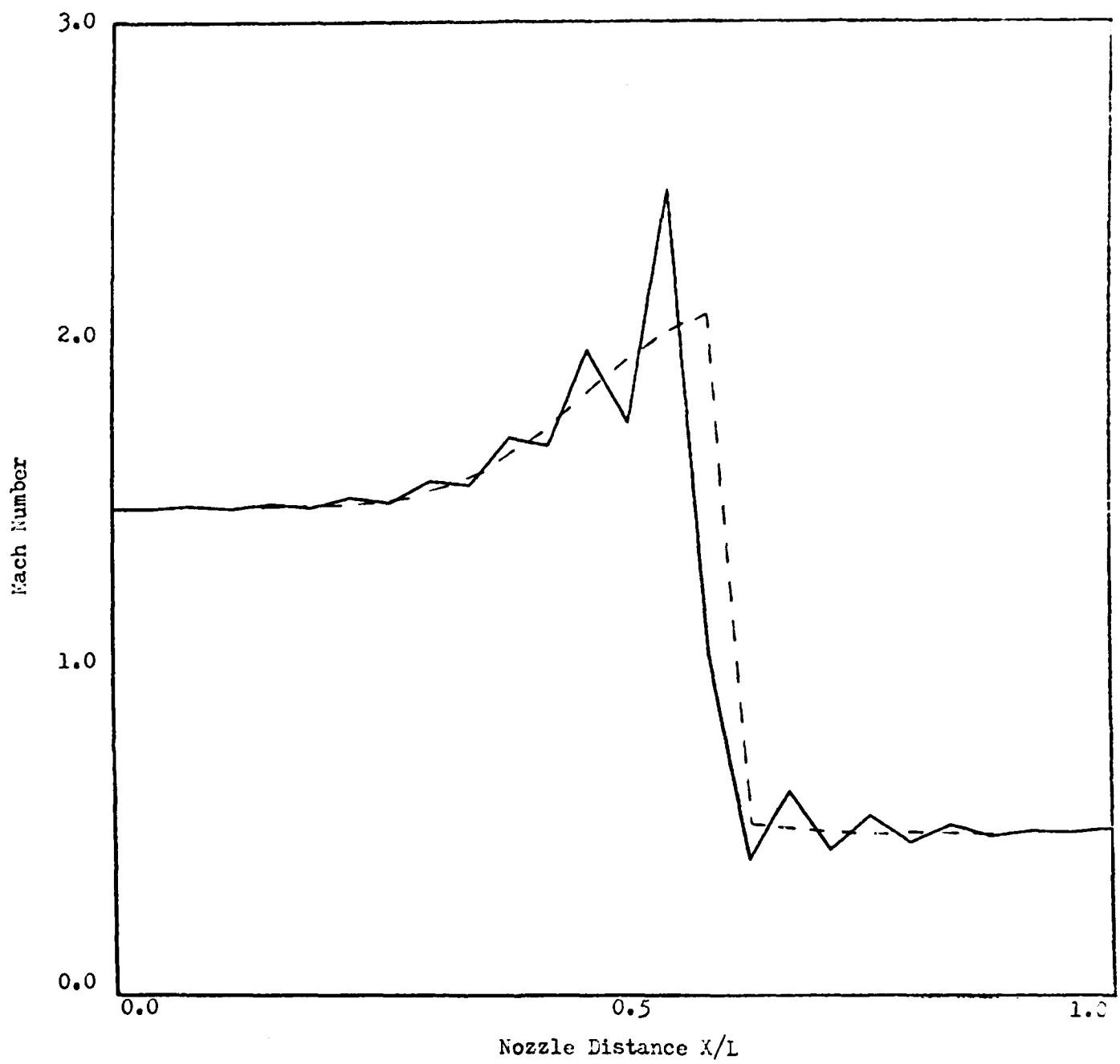
Scheme: Explicit
 Nozzle Area: $1.398 + 0.347 \operatorname{Tanh} 0.8(x-4)$
 B.C.: Specify ρ
 Theory: — — —
 Data Results: — — —

Figure 14. Mach Number Distribution for Subsonic Flow with a Shock



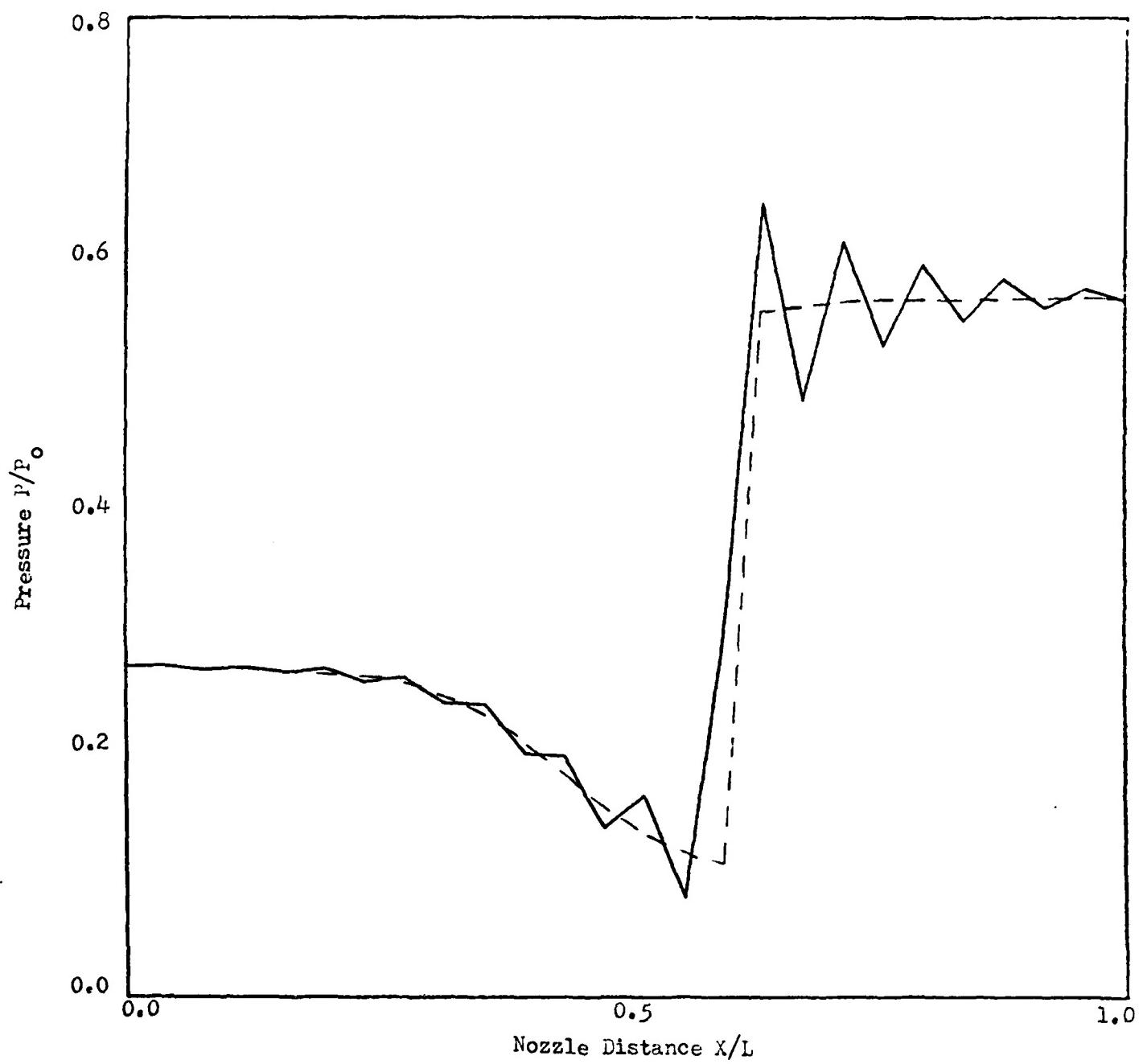
Scheme: Explicit
 Nozzle Area: $1.398 + 0.347 \operatorname{Tanh} 0.8(x-4)$
 B.C.: Specify P
 Theory: - - -
 Data Results: —

Figure 15. Pressure Distribution for Subsonic Flow with a Shock



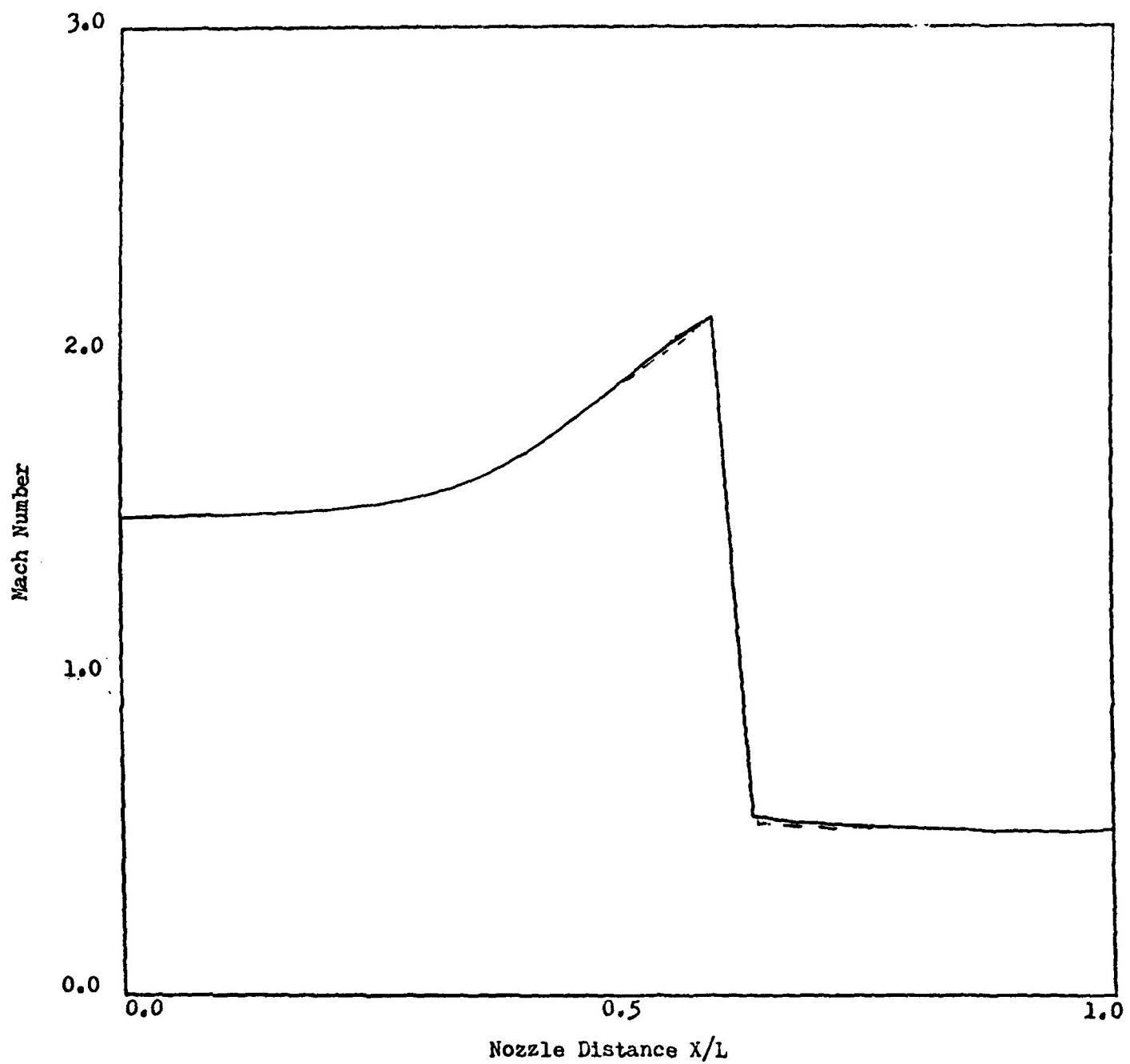
Scheme: Explicit
 Nozzle Area: $1.398 + 0.347 \tanh 0.8(x-4)$
 B.C.: Specify P_{∞}
 Theory: — — —
 Data Results: — — —

Figure 16. Mach Number Distribution for Subsonic Flow with a Shock



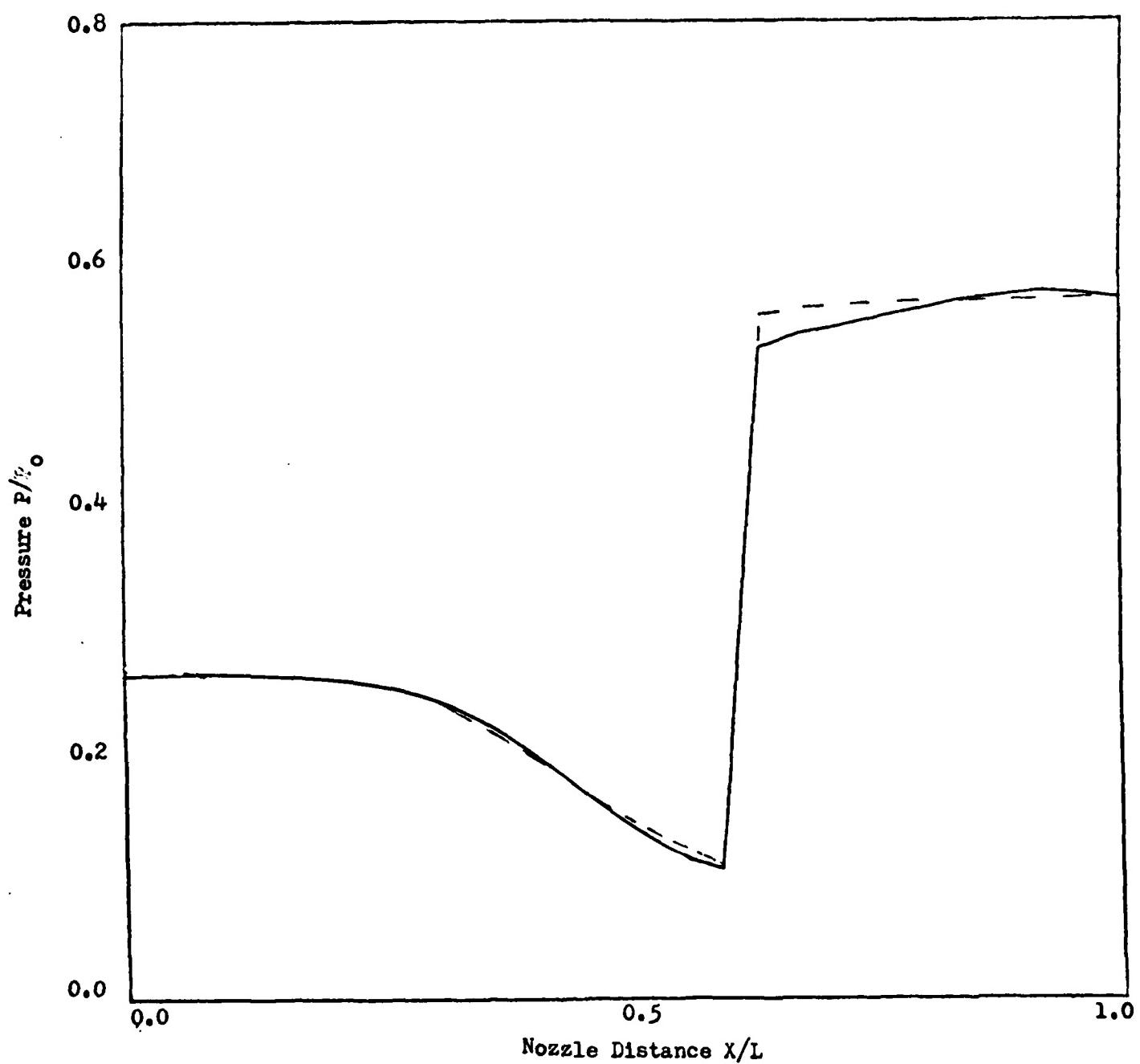
Scheme: Explicit
 Nozzle Area: $1.398 + 0.347 \operatorname{Tanh} 0.8(x-4)$
 B.C.: Specify P_∞
 Theory: ——
 Data Results: ———

Figure 17. Pressure Distribution for Subsonic Flow with a Shock



Scheme: Explicit
Nozzle Area: $1.398 + 0.347 \operatorname{Tanh} 0.8(x-4)$
B.C.: Specify P&U
Theory: — — —
Data Results: —————
t: Variable

Figure 18. Mach Number Distribution for Subsonic Flow with a Shock



Scheme: Explicit
 Nozzle Area: $1.398 + 0.347 \operatorname{Tanh} 0.8(x-4)$
 B.C.: Specify P
 Theory: — — —
 Data Results: ——————
 Δt : Variable

Figure 19. Pressure Distribution for Subsonic Flow with a Shock

because the truncation error from the time derivative term cancelled the artificial viscosity generated by the convective term. (For further information concerning this point see ref. 15).

The results up to this point showed that the affects of both boundary conditions are about the same.

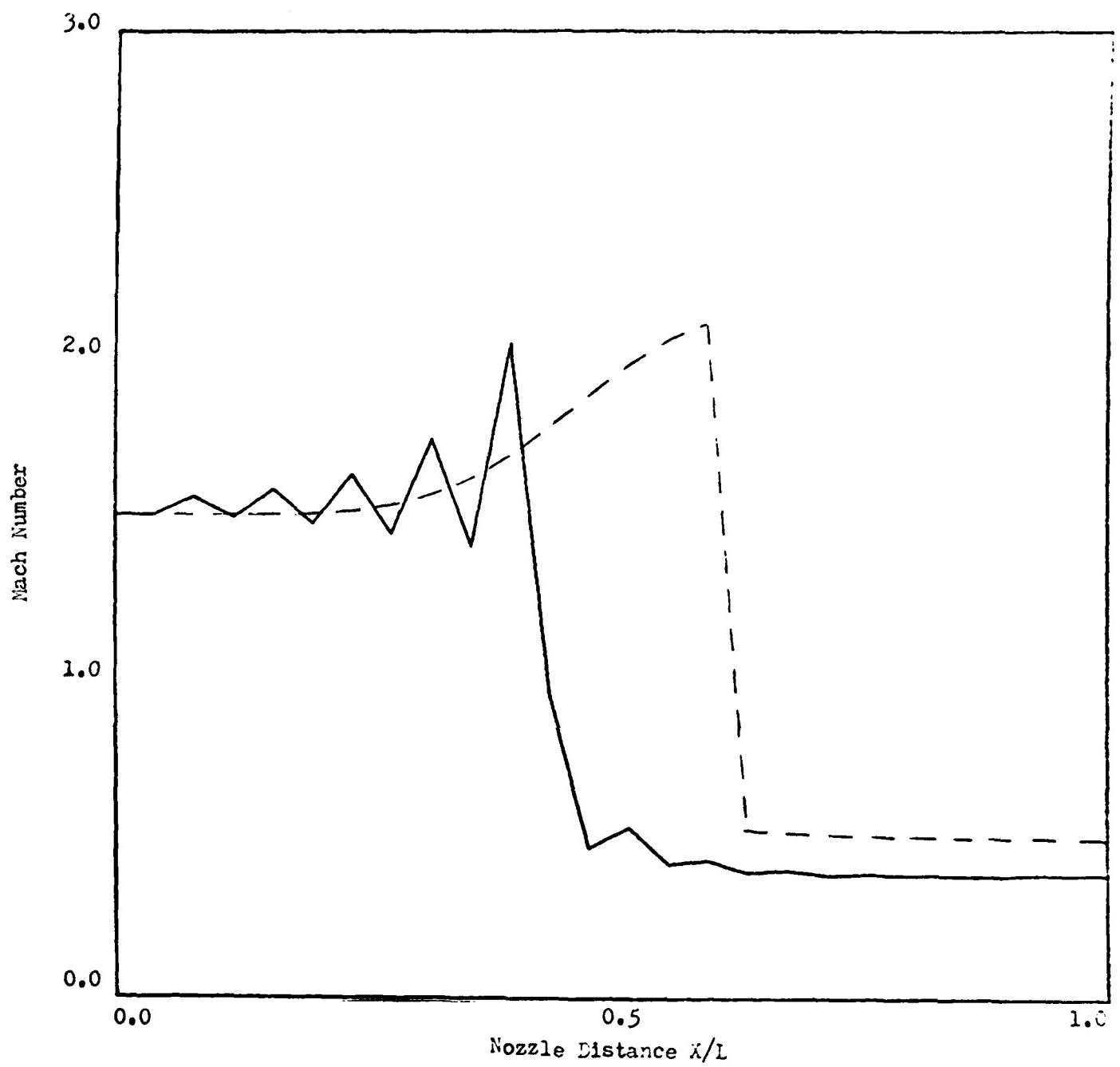
A special test was run to determine how MacCormack's algorithms could handle the relationship between shock location and back pressure. Both boundary conditions were used to test their effects on the algorithms. The results are located in Figures 20 through 27. The test was run by allowing the scheme to converge to an initial guess. Then, after convergence was achieved the boundary condition was changed. It was initially hoped that the implicit scheme would allow the shock to move upstream especially when McKenna's boundary condition was used. This condition was the one that specified pressure and velocity at the exit. It was thought that by specifying two variables at the boundary it would "force" the shock to move, but such was not the case.

The first test case involved the implicit algorithm. McKenna's boundary condition did not work for either moving the shock upstream or downstream. Immediately after the initial guess converged and the boundary condition changed, negative values for the pressure and density appeared. The results were meaningless. The results for the implicit algorithm with Steger's boundary conditions were the same as McKenna's. They both failed. They did the same thing--produced negative values for pressure and density and became numerically unstable. This was surprising. It was hoped that MacCormack's implicit algorithm could handle the problem of allowing the shock to move.

The results for the explicit case were successful. McKenna's boundary condition that specified pressure and velocity still did not allow the shock to move upstream or downstream, but Steger's boundary condition that specified just the pressure did. As Figures 20 through 23 show, Steger's boundary condition moves the shock up and down the nozzle right where it is supposed to be (for the corresponding pressure). This is an interesting point. It shows that the boundary condition that specifies only pressure is the logical choice to use both theoretically and numerically. Theoretically, pressure is the driving mechanism for nozzle flow problems. Numerically, specifying pressure at the boundary works.

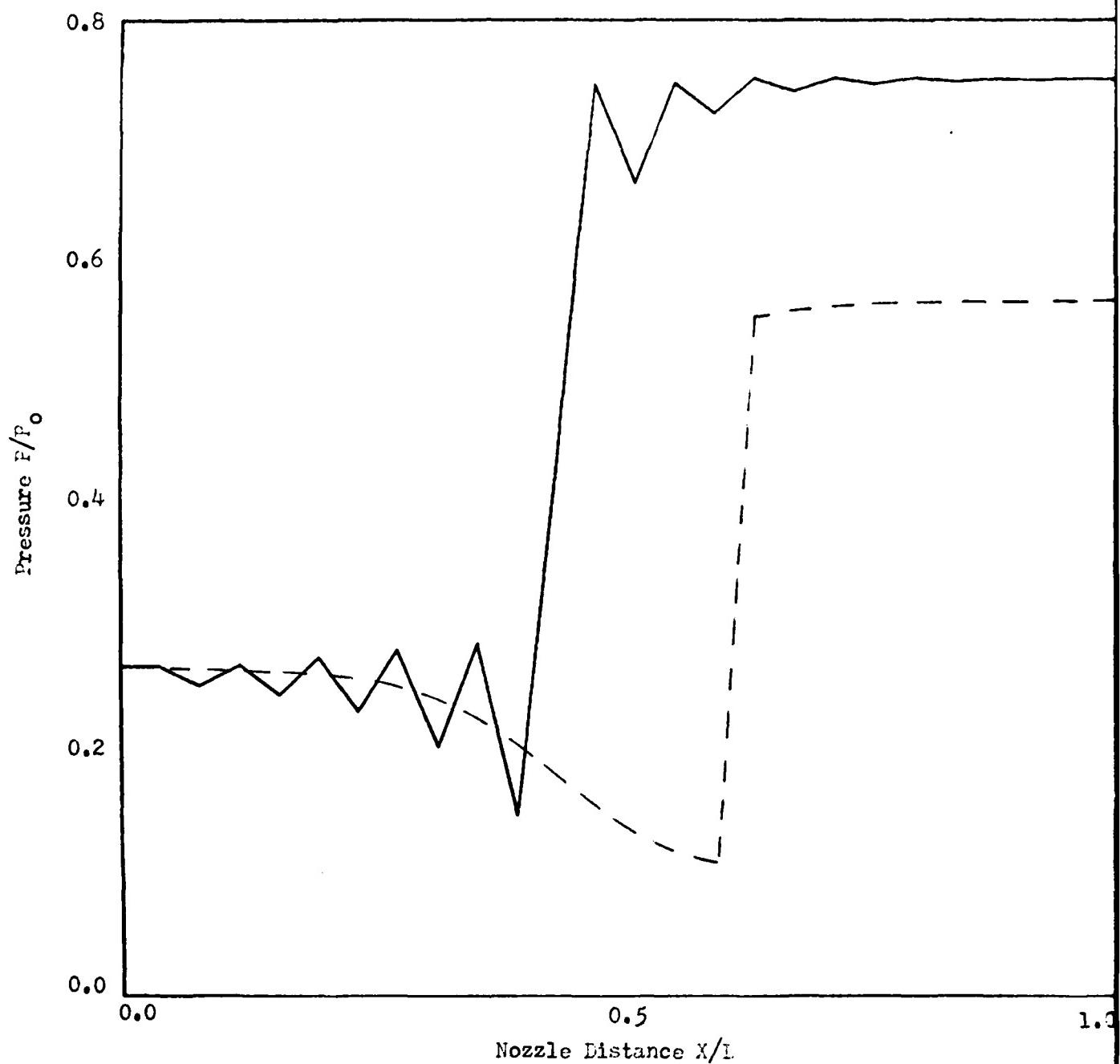
It was then decided to allow the time-step (Δt) to float to see if the jumps could be smoothed out. The test was run for both boundary conditions in the explicit algorithm. The results for attempting to move the shock upstream are located in Figures 24 and 25. Only Steger's boundary condition was stable. Notice, the shock did not move. In fact, the results are exactly the same from the inflow to the shock location (original location) as the initial theoretical values. Once past the shock the results for each boundary condition varied. Analyzing the results it appears as though once the shock initially forms, a change in the downstream boundary condition cannot be felt upstream of the shock. The effects of the downstream boundary conditions are just reflected back and forth from the shock to the exit.

The attempts to let the shock move downstream were disappointing. Neither boundary condition worked. The condition that specified only pressure did converge but again the shock did not move. McKenna's boundary condition was again unstable. The upstream values were the



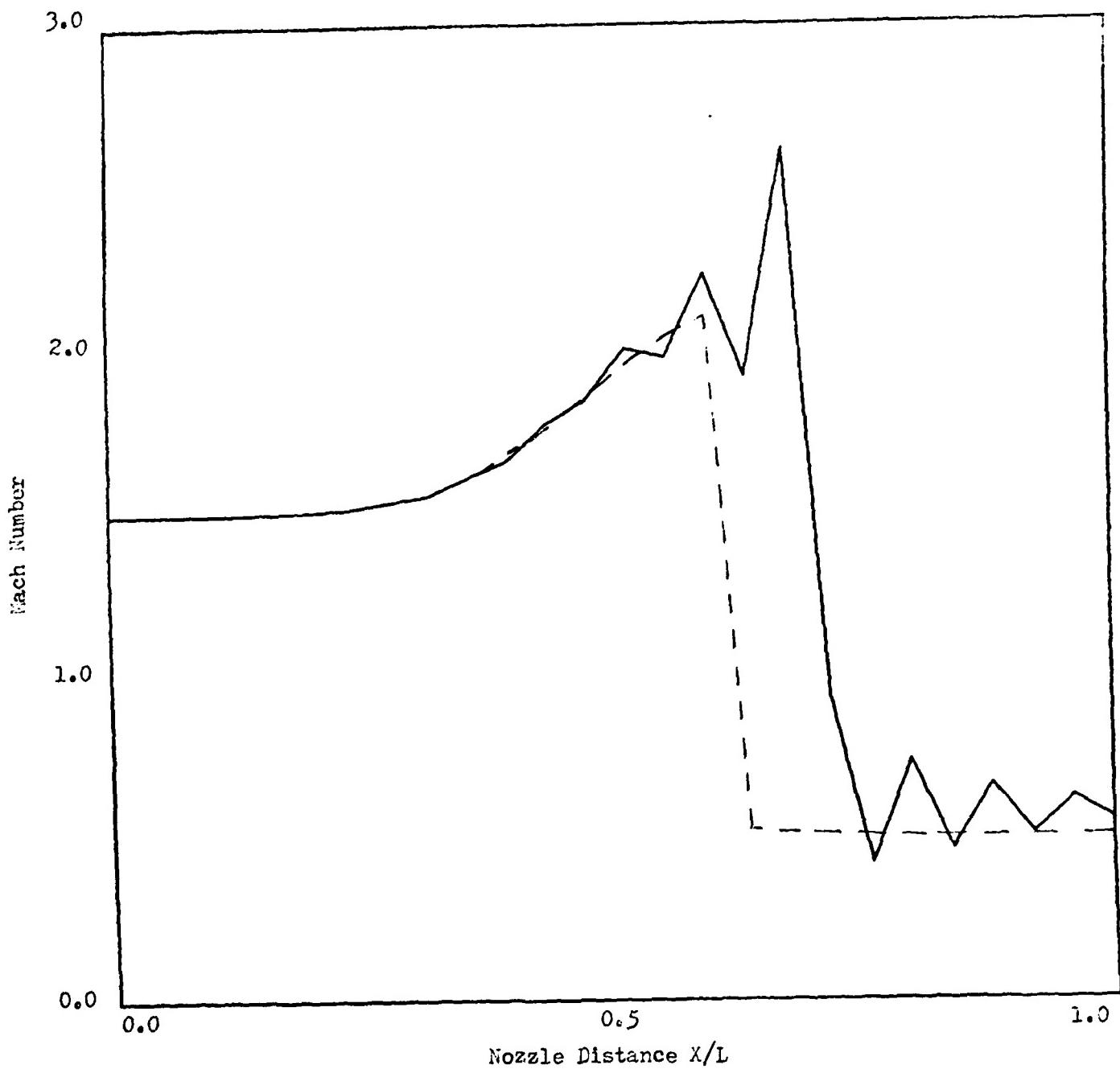
Scheme: Explicit
 Nozzle Area: $1.398 + 0.347 \operatorname{Tanh} 0.8(x-4)$
 B.C.: Specify P
 Original Shock: - - -
 Data Results: —————
 Shock Action: Moved Upstream

Figure 20. Mach Number Distribution for Moving the Shock Upstream



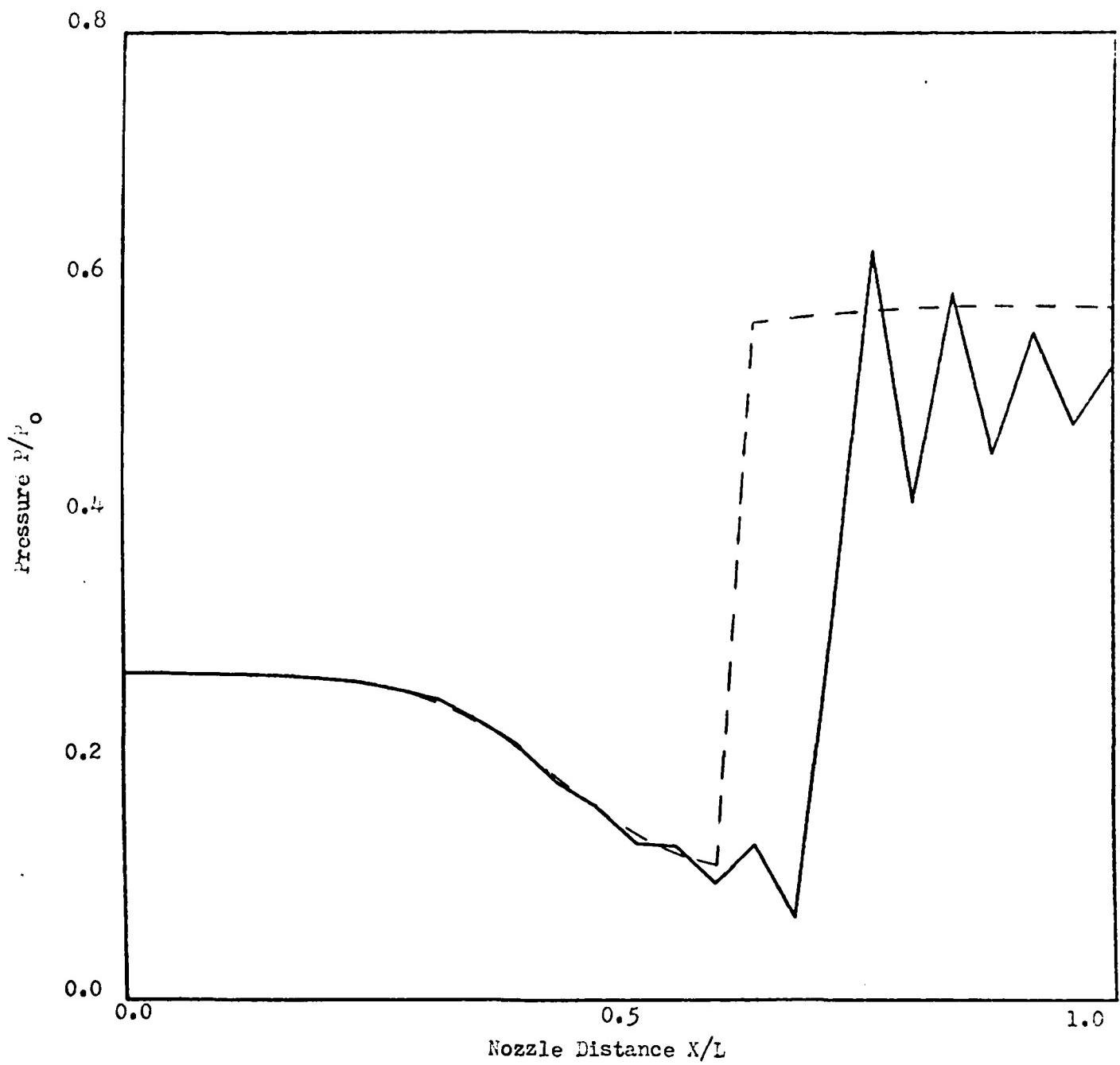
Scheme: Explicit
 Nozzle Area: $1.398 + 0.347 \operatorname{Tanh} 0.8(x-4)$
 B.C.: Specify P
 Original Shock: — — —
 Data Results: — — —
 Shock Action: Moved Upstream

Figure 21. Pressure Distribution for Moving the Shock Upstream



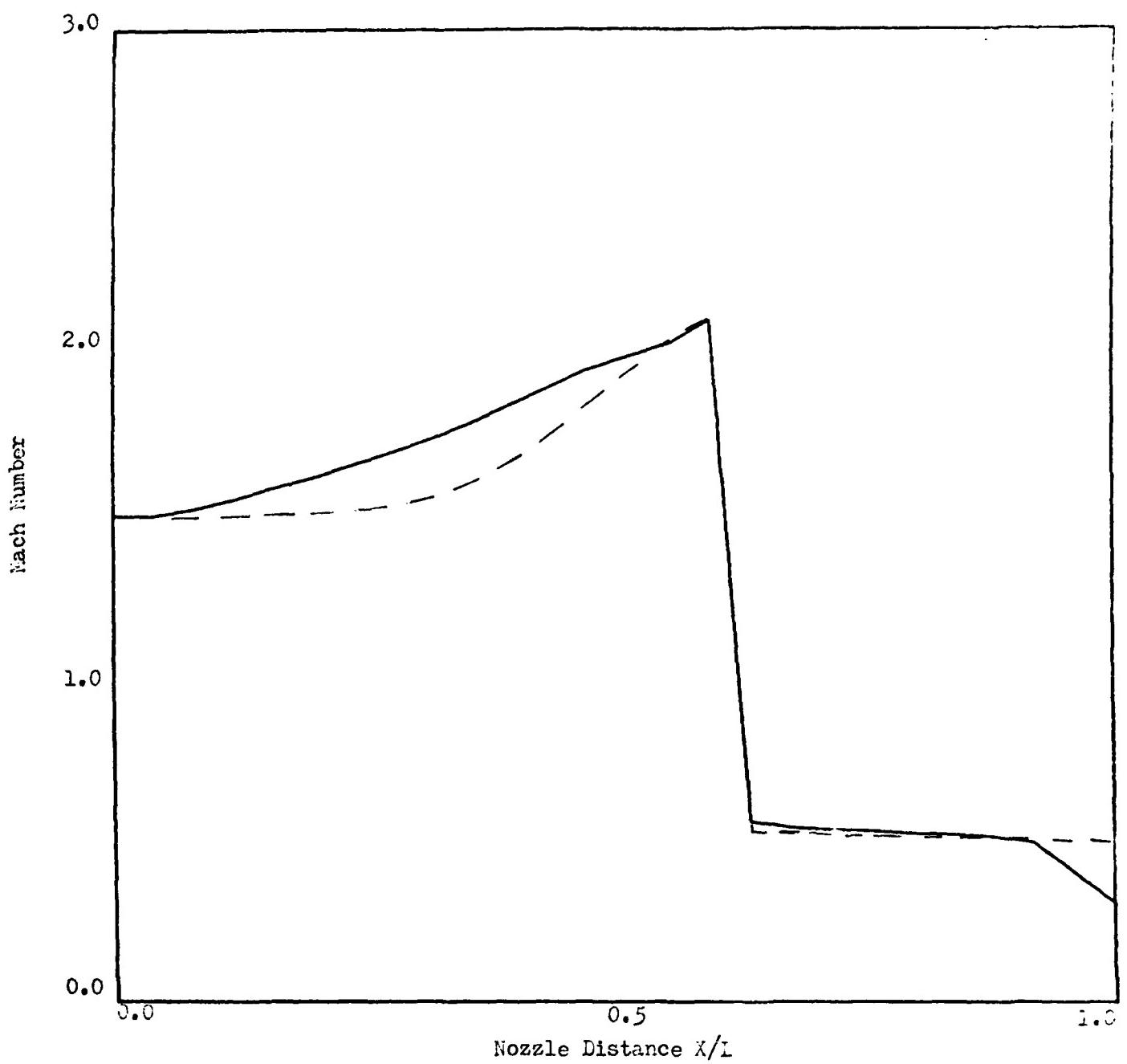
Scheme: Explicit
 Nozzle Area: $1.398 + 0.347 \operatorname{Tanh} 0.8(x-4)$
 B.C.: Specify P
 Original Shock: — — —
 Data Results: _____
 Shock Action: Moved Downstream

Figure 22. Mach Number Distribution for Moving the Shock Downstream



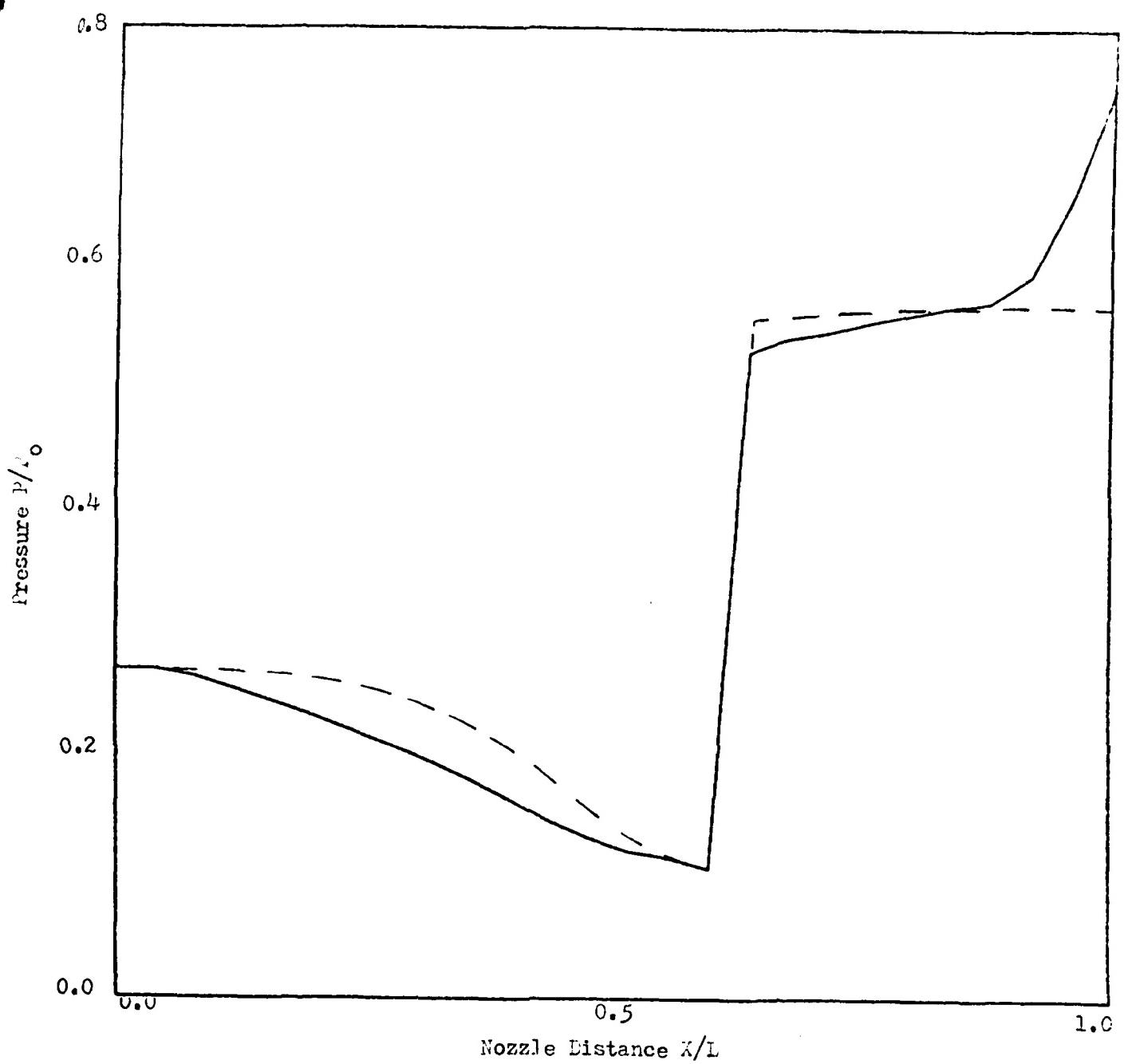
Scheme: Explicit
 Nozzle Area: $1.398 + 0.347 \operatorname{Tanh} 0.8(x-4)$
 B.C.: Specify P
 Original Shock: — — —
 Data Results: ——————
 Shock Action: Moved Downstream

Figure 23. Pressure Distribution for Moving the Shock Downstream



Scheme: Explicit
 Nozzle Area: $1.398 + 0.347 \operatorname{Tanh} 0.8(x-4)$
 B.C.: Specify P
 Original Shock: — — —
 Data Results: —————
 Δt : Variable
 Shock Action: Should Have Moved Upstream

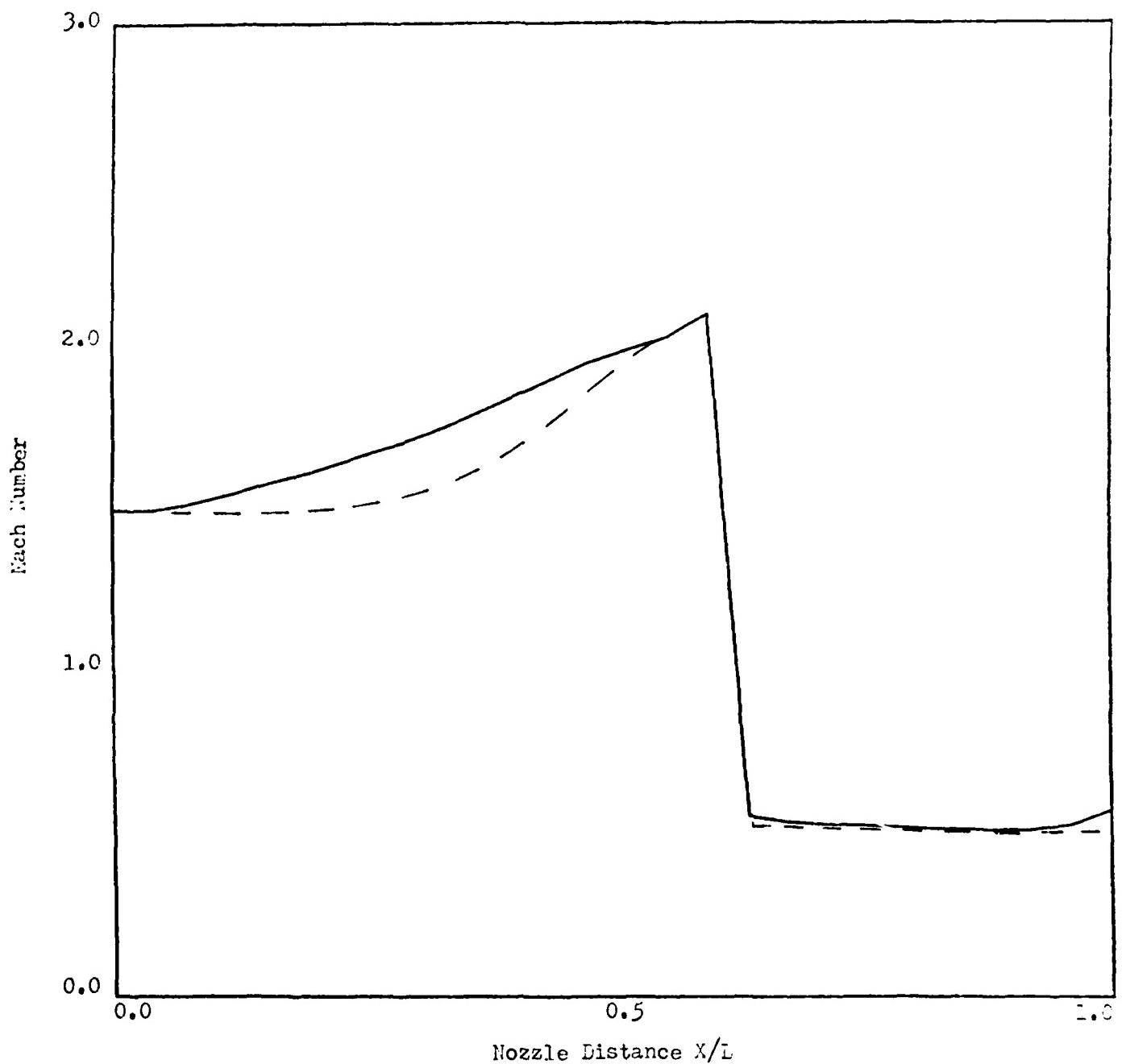
Figure 24. Mach Number Distribution for Trying to Move the Shock Upstream



Scheme: Explicit
 Nozzle Area: $1.398 + 0.347 \operatorname{Tanh} 0.8(x-4)$
 B.C.: Specify P
 Original Shock: — — —
 Data Results: ——————
 Δt : Variable
 Shock Action: Should Have Moved Upstream

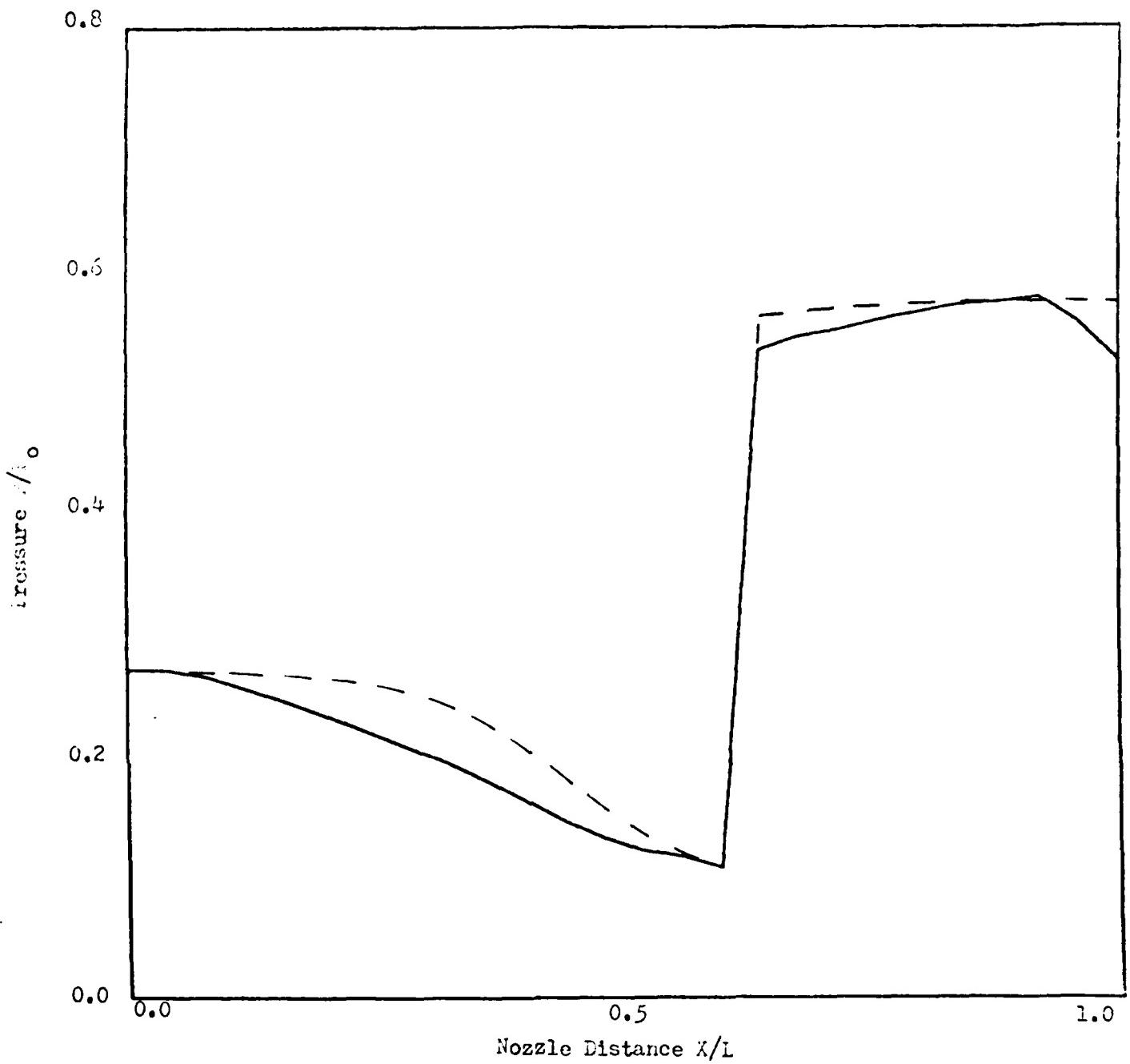
Figure 25. Pressure Distribution for Trying to Move the Shock Upstream

same as the initial guess (converged solution), but the downstream values were radically different (same as before). See Figures 26 and 27 for the results. The results for this section were very surprising. It was thought that the shock would move downstream (maybe not upstream, but at least downstream) due to the subsonic nature of the flow, but as stated, the results readily proved this inaccurate. This is an interesting case because when Δt is held constant the shock moves!



Scheme: Explicit
 Nozzle Area: $1.398 + 0.347 \operatorname{Tanh} 0.8(x-4)$
 B.C.: Specify P
 Original Shock: — — —
 Data Results: ——————
 Δt : Variable
 Shock Action: Should Have Moved Downstream

Figure 26. Mach Number Distribution for Trying to Move the Shock Downstream



Scheme: Explicit
 Nozzle Area: $1.398 + 0.347 \operatorname{Tanh} 0.8(x-4)$
 B.C.: Specify P
 Original Shock: — — —
 Data Results: — — —
 Δt : Variable
 Shock Action: Should Have Moved Downstream

Figure 27. Pressure Distribution for Trying to Move the Shock Downstream

VI. CONCLUSIONS

Numerical experiments for the quasi-one-dimensional Euler equations have been applied to nozzle flow problems. These experiments included boundary condition analysis and a comparison of MacCormack's implicit and explicit algorithms. The results from these experiments are judged by how well the equations are solved. The judgment criteria is convergence rate, robustness, and adaptability.

There were two (exit) boundary conditions used, each based on characteristic theory. One boundary condition specified pressure and velocity (characteristic variable) while the other only specified pressure. The convergence rate for both conditions was just about the same. The condition that specified pressure and velocity was initially thought to be mathematically superior in that a non-reflecting boundary condition would prevail. The results showed just the opposite. The condition that only specified pressure was robust. This was the only boundary condition in which the shock moved upstream and downstream. The adaptability of both boundary conditions are about the same. This is a result of the fact that they were both derived from characteristic theory. Judging the two boundary conditions, it is evident that the condition which specifies only pressure is more useful.

Each of MacCormack's algorithms has their advantages and disadvantages. The explicit method is very reliable and can be used for a wide variety of problems. The main drawback is the large amount of computing time required to solve a problem. MacCormack's implicit

method is a new method which yields highly accurate results with a considerable amount of reduction in computer time. The method is not truly implicit because it is not unconditionally stable. The main advantage of the implicit method is that when the problem becomes more complex (viscous effects and turbulence) the explicit part of the scheme becomes more complex too, but the implicit numerics are not affected--therefore, a great reduction in computer time exists. The inability of the implicit scheme to move the shock upstream or downstream was a disappointment.

Areas of further study that need to be investigated are:

- 1) McKenna's boundary condition and why they did not work for the explicit case. 2) The importance of initial conditions. (This might account for the fact that McKenna's boundary condition did not work.)

BIBLIOGRAPHY

- ¹Richtmeyer, R. D. and Morton, K. W., Difference Methods for Initial-Value Problems, 2nd ed., Interscience Publishers, New York, 1967.
- ²MacCormack, R. W. and Lomax, H., "Numerical Solution of Compressible Viscous Flows," Ann. Rev. Fluid Mech., 11, 1979, pp. 289-316.
- ³Hollanders, H. and Viviand, H., "The Numerical Treatment of Compressible High Reynolds Number Flows," Von Karman Institute for Fluid Dynamics, Lecture Series 1979-6, 1979.
- ⁴Kreiss, H. O., "Initial Boundary Value Problem for Hyperbolic Systems," Communications on Pure and Applied Mathematics, Vol. 23, 1970, pp. 277-298.
- ⁵Pulliam, T. H., "Characteristic Boundary Conditions For the Euler Equations," Numerical Boundary Condition Procedures, Proceedings of a Symposium held at NASA Ames Research Center, Moffett Field, California, Oct. 19-20, 1981.
- ⁶Yee, H. C., Beam, R. M. and Warming, R. F., "Stable Boundary Approximations For A Class of Implicit Schemes For The One-Dimensional Inviscid Equations of Gas Dynamics," AIAA paper 81-1009, AIAA Computational Fluid Dynamics Conference, Palo Alto, California, June 22-23, 1981.
- ⁷Bell, J. B., Shubin, G. R., and Solomon, J. M., "Fully Implicit Shock Trucking," Numerical Boundary Condition Procedures, Proceedings of a Symposium held at NASA Ames Research Center, Moffett Field, California, Oct. 19-20, 1981.
- ⁸Chakravarthy, S. R., "Euler Equations - Implicit Schemes and Implicit Boundary Conditions," AIAA paper 82-0228 AIAA Aerospace Sciences Meeting, Orlando, Florida, Jan. 11-14, 1982.
- ⁹Chakravarthy, S. R., Anderson, D. A. and Salas, M. D., "The Split-Coefficient Matrix Method for Hyperbolic Systems of Gas Dynamic Equations," AIAA paper 80-0268 AIAA Aerospace Sciences Meeting, Pasadena, California, Jan. 14-16, 1980.
- ¹⁰MacCormack, R. W., "The Effect of Viscosity on Hypervelocity Impact Cratering," AIAA paper 690-354, AIAA Hypervelocity Impact Conference, Cincinnati, Ohio, Apr. 30-May 2, 1969.

¹¹ MacCormack, R. W., "A Numerical Method for Solving the Equations of Compressible Viscous Flow," AIAA paper 81-0110, AIAA 19th Aerospace Sciences Meeting, St. Louis, Missouri, Jan. 12-15, 1981.

¹² White, M. E., and Anderson, J. D., Jr., "Application of Mac-Cormack's Implicit Method to Quasi-One-Dimensional Nozzle Flows," AIAA paper 82-0992, AIAA/ASME 3rd Joint Thermophysics, Fluids, Plasma and Heat Transfer Conference, St. Louis, Missouri, Jun 7-11, 1982.

¹³ McKenna, P. J., Graham, J. E., and Hankey, W. L., "The Role of Far-Field Boundary Conditions in Numerical Solutions of the Navier-Stokes Equations," AFWAL-TR-82-3029.

¹⁴ Steger, J. L., Pulliam, T. H., and Chima, R. V., "An Implicit Finite Difference Code for Inviscid and Viscous Cascade Flow," AIAA Preprint-80-1427, AIAA 13th Fluid and Plasma Dynamics Conference, July 14-16, 1980.

¹⁵ Hodge, J. K., and Hankey, W. L., notes on "Implicit Methods for Model Equations."

APPENDIX A

Derivation of the Jacobian

The Jacobian (A) is defined as

$$A_{1m}(U) = \frac{\partial F_1}{\partial U_m}$$

In order to calculate it, the flux vector "F" should be redefined in terms of the conservative variables "U".

Given

$$U = \begin{Bmatrix} \rho \\ \rho u \\ e \end{Bmatrix} = \begin{Bmatrix} \rho \\ m \\ e \end{Bmatrix}$$

then "F" can be rewritten as

$$F = \begin{Bmatrix} \rho u \\ p + \rho u^2 \\ u(e + p) \end{Bmatrix} = \begin{Bmatrix} m \\ (\gamma - 1)e + \frac{3-\gamma}{2} \frac{m^2}{\rho} \\ \frac{\gamma me}{\rho} - \frac{(\gamma - 1)}{2} \frac{m^3}{\rho^2} \end{Bmatrix}$$

The Jacobian can now be computed as

$$A_{11} = \frac{\partial F_1}{\partial U_1} = \frac{\partial m}{\partial \rho} = 0$$

$$A_{12} = \frac{\partial F_1}{\partial U_2} = \frac{\partial m}{\partial m} = 1$$

$$A_{13} = \frac{\partial F_1}{\partial U_3} = \frac{\partial m}{\partial e} = 0$$

$$A_{21} = \frac{\partial F_2}{\partial U_1} = \frac{\partial \left((\gamma-1) e + \left(\frac{3-\gamma}{2}\right) \frac{m^2}{\rho} \right)}{\partial \rho} = -\frac{(3-\gamma)}{2} \frac{m^2}{\rho^2}$$

$$A_{22} = \frac{\partial F_2}{\partial U_2} = \frac{\partial \left((\gamma-1) e + \left(\frac{3-\gamma}{2}\right) \frac{m^2}{\rho} \right)}{\partial m} = (3-\gamma) \frac{m}{\rho}$$

$$A_{23} = \frac{\partial F_2}{\partial U_3} = \frac{\partial \left((\gamma-1) e + \left(\frac{3-\gamma}{2}\right) \frac{m^2}{\rho} \right)}{\partial e} = (\gamma-1)$$

$$A_{31} = \frac{F_3}{U_1} = \frac{\partial \left(\frac{\gamma m e}{\rho} - \left(\frac{\gamma-1}{2}\right) \frac{m^3}{\rho^2} \right)}{\partial \rho} = -\frac{\gamma m e}{\rho^2} + \frac{(\gamma-1)m^3}{\rho^3}$$

$$A_{32} = \frac{\partial F_3}{\partial U_2} = \frac{\partial \left(\frac{\gamma m e}{\rho} - \left(\frac{\gamma-1}{2}\right) \frac{m^3}{\rho^2} \right)}{\partial m} = \frac{\gamma e}{\rho} - \frac{3}{2} (\gamma-1) \frac{m^2}{\rho^2}$$

$$A_{33} = \frac{\partial F_3}{\partial U_3} = \frac{\partial \left(\frac{\gamma m e}{\rho} - \left(\frac{\gamma-1}{2}\right) \frac{m^3}{\rho^2} \right)}{\partial e} = \frac{\gamma m}{\rho}$$

Summarizing we have

$$A = \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix}$$

or

$$A = \begin{bmatrix} 0 & 1 & 0 \\ -\frac{3-\gamma}{2} \frac{m^2}{\rho^2} & (3-\gamma) \frac{m}{\rho} & (\gamma-1) \\ -\frac{\gamma m e}{\rho^2} + (\gamma-1) \frac{m^3}{\rho^3} & \frac{\gamma e}{\rho} - \frac{3}{2} (\gamma-1) \frac{m^2}{\rho^2} & \frac{\gamma m}{\rho} \end{bmatrix}$$

APPENDIX B

Characteristic Boundary Conditions: P & U Specified (McKenna)

The characteristic variables are

$$Q_1 = \rho - p/C_0^2$$

$$Q_2 = p + \rho_0 C_0 U$$

$$Q_3 = p - \rho_0 C_0 U$$

In terms of primitive variables they are

$$p = \frac{Q_2 + Q_3}{2}$$

$$U = \frac{Q_2 - Q_3}{2\rho_0 C_0}$$

$$\rho = Q_1 + \frac{Q_2 + Q_3}{2C_0^2}$$

These values were derived by solving the three equations (above) for three unknowns.

This information is now applied at the subsonic (exit) boundary.

Subsonic Outflow:

The boundary conditions are mathematically represented as

$$\frac{\partial Q_1}{\partial x} = 0$$

$$\frac{\partial Q_2}{\partial x} = 0$$

$$Q_3 = \text{constant} = K_3$$

Refer now to the primitive variables written in terms of the characteristics. The velocity, for example is written as

$$U = \frac{Q_2 - Q_3}{2\rho_0 C_0}$$

This term contains two characteristics - the second and third. McKenna reasoned that Q_3 is constant while Q_1 and Q_2 (at the exit) are allowed to float. Therefore in order to set the exit boundary condition, Q_1 and Q_2 should be expressed in terms of the primitive variables. Q_3 is held constant at K_3 . The velocity term would then become

$$U = \frac{\frac{Q_2}{p + \rho_0 C_0 U} - K_3}{2\rho_0 C_0}$$

The other terms are

$$\rho = (\rho - \frac{p}{C_0^2}) + \frac{(p + \rho_0 C_0 U)}{2C_0^2} + K_3$$

$$p = \frac{(p + \rho_0 C_0 U)}{2} + K_3$$

The variables are numerically implemented into the program as:

$$P_j^n = \frac{\rho_0 C_0}{2} \left[\frac{P_{j-1}^n}{\rho_0 C_0} + U_{j-1}^n + \frac{K_3}{\rho_0 C_0} \right]$$

$$U_j^n = \frac{1}{2} \left[U_{j-1}^n + \frac{P_{j-1}^n}{\rho_0 C_0} - \frac{K_3}{\rho_0 C_0} \right]$$

$$\rho_j^n = \rho_{j-1}^n + \frac{\rho_0}{2C_0} \left[\frac{K_3}{\rho_0 C_0} - \frac{P_{j-1}^n}{\rho_0 C_0} + U_{j-1}^n \right]$$

Where ρ_0 & C_0 are evaluated at the j^{th} point for the $n - 1$ time-step.

K_3 is a constant represented as

$$K_3 = P_\infty - C_0 \rho_0 U_\infty$$

Here P_∞ and U_∞ are the values at the exit.

APPENDIX C

Characteristic Boundary Condition: P specified (Steger)

Subsonic Outflow:

Steger represented the outflow boundary as

$$\frac{\partial Q_1}{\partial x} = 0$$

$$\frac{\partial Q_2}{\partial x} = 0$$

$$P = P_\infty$$

This can be rewritten as

$$Q_{1,j} = Q_{1,j-1}$$

$$Q_{2,j} = Q_{2,j-1}$$

$$P = P_\infty$$

In terms of the primitive variables they are written as

$$\rho_j - P_j/C_0^2 = \rho_{j-1} - P_{j-1}/C_0^2$$

$$P_j + \rho_0 C_0 U_j = P_{j-1} + \rho_0 C_0 U_{j-1}$$

$$P_j = P_\infty$$

Now solving for the density and velocity we get

$$\rho_j = \rho_{j-1} + \frac{1}{C_0^2} (P_\infty - P_{j-1})$$

$$u_j = u_{j-1} + \frac{1}{\rho_0 C_0} (P_{j-1} - P_\infty)$$

$$P_j = P_\infty$$

Numerically these equations are represented as

$$\rho_j^n = \rho_{j-1}^n + \frac{1}{C_0^2} (P_\infty - P_{j-1}^n)$$

$$u_j^n = u_{j-1}^n + \frac{1}{\rho_0 C_0} (P_{j-1}^n - P_\infty)$$

$$P_j^n = P_\infty$$

ρ_0 and C_0 are the same as before.

APPENDIX D

Sample Calculations for the Initial Shock Location:

$$P_0 = 10,000 \text{ lb}_f/\text{ft}^2$$

Given:

$$\rho_0 = .00883 \text{ lb}_f \text{ sec}^2 \text{ ft}^4$$

The values of various points are:

PT #2

$$(x = .4) \quad A_2 = 1.0514 \quad M_2 = 1.5$$

$$\frac{A}{A^*} = 1.176 \quad \frac{P}{P_0} = .2724 \quad \frac{\rho}{\rho_0} = .3950$$

$$A^* = .8941 \quad P_2 = 2724 \quad \rho_2 = .0034 \quad C = 1045.6$$

$$U_2 = 1568.45$$

PT #1

$$(x = 0.0) \quad \frac{A_1}{A^*} = \frac{1.0512}{.8941} = 1.1757 \quad M = 1.4996$$

$$\frac{P}{P_0} = .2726 \quad \frac{\rho}{\rho_0} = .3951 \quad C = 1045.9$$

$$U_1 = 1568.38 \quad P_1 = 2726 \quad \rho_1 = .0034$$

The shock is at point 16:

PT #16

$$(X = 6.0) \quad \frac{A_{16}}{A^*} = \frac{1.6284}{.8941} = 1.821 \quad M_{16} = 2.09$$

$$\frac{P}{P_0} = .1111 \quad \frac{\rho}{\rho_0} = .2081 \quad C = 920.04$$

$$U_{16} = 1922.9 \quad P_{16} = 1111 \quad \rho_{16} = .0018$$

After the shock

$$M_y = .5628 \quad \frac{P_2}{P_1} = 4.929 \quad \frac{\rho_2}{C_1} = 2.798$$

$$\frac{P_{0y}}{P_{0x}} = .6789 \quad \frac{P_{16}}{P_{02}} = .1636$$

$$U_y = 687.3 \quad P_y = 5476.12 \quad \rho_y = .0051$$

$$\frac{P}{P_{0y}} = .8065 \quad \frac{\rho}{\rho_{0y}} = .8576$$

$$P_{0y} = 6789 \quad \rho_{0y} = .0059 \quad A_y = 1.2363$$

$$\frac{A_E}{A_E} = \frac{A_{25}}{A^*} \quad \frac{A^*}{A_{16}} \quad \frac{A_{16}}{A_y} \quad \frac{A_y}{A_y} = \frac{A_{25}}{A_y}$$

$$\frac{A_E}{A_E} = \frac{1.7445}{.8941} \cdot \frac{1}{1.821} \cdot \frac{1}{1} \cdot \frac{1.2363}{1} = 1.3248$$

$$M_{25} = .508 \quad \frac{P}{P_{0y}} = .8385 \quad \frac{\rho}{\rho_{0y}} = .8817 \quad C_{25} = 1228.6$$

PT #25

$$(X = 9.6) \quad U_{25} = 624 \quad P_{25} = 5692.6 \quad \rho_{25} = .0052$$

PT #26

$$(X = 10.0) \quad \frac{A_E}{A_E} = \frac{1.7447}{.8941} \cdot \frac{1}{1.821} \cdot \frac{1.2363}{1} = 1.3249 \quad M_{26} = .508$$

(same as PT #25)

| <u>PT</u> | <u>U</u> | <u>P</u> | <u>ρ</u> | <u>M</u> |
|-----------|----------|----------|----------|----------|
| 1 | 1568.38 | 2726 | .0034 | 1.4996 |
| 2 | 1568.45 | 2724 | .0034 | 1.5 |
| 16 | 1922.9 | 1111 | .0018 | 2.09 |
| y | 687.3 | 5476.12 | .0051 | .5628 |
| 25 | 623.99 | 5692.6 | .0052 | .508 |
| 26 | 623.0 | 5692.9 | .0052 | .5080 |

APPENDIX E

Theoretical Values for Moving the Shock Upstream and Downstream:

Upstream: (From the original shock) Choose point #12 for the new shock location

$$PT \#12 \quad \frac{A_{12}}{A^*} = \frac{1.2431}{.8941} = 1.3904 \quad M_{12} = 1.7540$$

$$\frac{P}{P_t} = .1867 \quad \frac{\rho}{\rho_t} = .3016$$

$$P_{12} = 1867 \quad = .0027 \quad U = 1725.8$$

$$M_y = .6271 \quad \frac{P_y}{P_{12}} = 3.422 \quad \frac{P_{ty}}{P_{t12}} = .8328 \quad \frac{P_y}{P_{12}} = 2.285 \quad \frac{P}{P_t} = .8276$$

$$P_y = 6388.9 \quad \rho_y = .0063 \quad U_y = 753.2 \quad P_{ty} = 8328 \quad \rho_{ty} = .0075$$

The exit conditions are now calculated

$$\frac{A_E}{A_E^*} = \frac{1.7445}{.8941} \cdot \frac{1}{1.3904} \cdot \frac{1}{1} \cdot \frac{1.1582}{1} = 1.6253$$

$$M_{25} = .3895 \quad \frac{P}{P_t} = .9007 \quad \frac{\rho}{\rho_t} = .9280$$

$$P_{25} = 7501 \quad \rho_{25} = .0070 \quad U_{25} = 477.1$$

Downstream: (From the original shock) Choose point #20 for the new downstream shock

$$PT \#20 \quad \frac{A_{20}}{A^*} = \frac{1.7343}{.8941} = 1.9398 \quad M_{20} = 2.1827 \quad M_y = .5521$$

$$\frac{\rho}{P_t} = .1919 \quad \frac{P_{t2}}{P_{t1}} = .6408 \quad \rho_{20} = .0017 \quad \frac{P_2}{P_1} = 2.8997$$

$$M = .5521 \quad \frac{A}{A^*} = 1.2519 \quad \frac{\rho}{\rho_t} = .8625 \quad P_{ty} = 6408 \quad \rho_{ty} = .002$$

The exit conditions are now calculated

$$\frac{A_E}{A_E^*} = \frac{1.7445}{.8941} \cdot \frac{1}{1.9398} \cdot \frac{1}{1} \cdot \frac{1.2519}{1} = 1.2593$$

$$M_{25} = .5472 \quad \frac{P}{P_t} = .8159 \quad \frac{\rho}{\rho_t} = .8647$$

$$P_{25} = 5228 \quad \rho_{25} = .0017 \quad U_{25} = 1,135$$

APPENDIX F - Program Listing

```

730.          DU(3,N)=0.
740.          ***** NOW RUN MACCORMACKS ALGORITHM *****
750.          ***** THIS IS THE PREDICTOR *****
760.          CALL SECOND(CPI)
770.          ***** THIS IS THE PREDICTOR *****
780.          ***** THIS IS THE PREDICTOR *****
790.          ***** THIS IS THE PREDICTOR *****
800.          DO 999 L=1,MAX
810.          DO 40 I=1,N
820.          C(I)=SORT GAMMA(U(3,I)/U(1,I))
830.          CONTINUE
840.          ***** THIS IS THE PREDICTOR *****
850.          CALL FILISX(U,C,N,S,ALF-1)
860.          CALL FILISX(U,C,N,SK)
870.          CALL FILUD(U,N,C,SIG,D)
880.          CALL FILUD(U,D,SIG,D1)
890.          CALL URUFIX(U,AR,UR,UF,N)
900.          M=M1
910.          CALL MATRX(SX,DU,U1,M+1)
920.          CALL MATXD(W1,U2,U3,M+1)
930.          CALL MATXS(U2,U3,M+1)
940.          CALL URUFIX(U(2,M),XC(M))
950.          DELU(1,M)-DT/DXX(UF(1,M+1)-UF(1,M))
960.          DELU(2,M)-DT/DXX(UF(2,M+1)-UF(2,M))
970.          *DT/DXX(U(3,M)*(AR(M+1)-ARM))
980.          DELU(3,M)-DT/DXX(UF(3,M+1)-UF(3,M))
990.          DO 60 I=1,3
1000.         U4(I,M)=DELU(I,M)*DT/DXX(U3,I,M+1)
1010.         CONTINUE
1020.         ***** THIS IS THE CORRECTOR *****
1030.         CALL MATRX(SX,W4,W5,M)
1040.         CALL MATDX(D1,W5,W6,M)
1050.         CALL MATXS(W6,DU,M)
1060.         CALL MATXD(W6,W7,M)
1070.         CALL MATXS(W7,U3,M)
1080.         IF(P.EQ.0) GO TO 70
1090.         M=P
1100.         GO TO 50
1110.         CONTINUE
1120.         DO 90 J=1,N
1130.         DO 80 I=1,3
1140.         UP(I,J)*UR(I,J)+DU(I,J)
1150.         CONTINUE
1160.         ***** NOW SOLVE FOR THE CORRECTOR *****
1170.         ***** NOW SOLVE FOR THE CORRECTOR *****
1180.         ***** NOW SOLVE FOR THE CORRECTOR *****
1190.         DO 100 I=1,N
1200.         UPX(1,I)*UP(1,I)/AR(I)
1210.         UPX(2,I)*UP(2,I)/(AR(I))UPX(1,I)
1220.         UPX(3,I)*UP(3,I)*0.4/ARM1-UPX(1,I)
1230.         *UPX(2,I)*0.2
1240.         CONTINUE
1250.         UPX(1,I)*U(1,I)
1260.         UPX(2,I)*U(2,I)
1270.         UPX(3,I)*U(3,I)
1280.         UPX(1,I)*U(1,2)
1290.         UPX(2,I)*U(2,2)
1300.         UPX(3,I)*U(3,2)
1310.         ...
1320.         ***** NOW SET THE PRIMITIVE VARIABLES *****
1330.         U(1,1)=0.003489
1340.         U(2,1)=1.56838
1350.         U(3,1)=2726.0
1360.         DO 30 I=2,16
1370.         U(1,I)=0.000118*(I)**0.00372
1380.         U(2,I)=25.32*(I**1517.8
1390.         U(3,I)=115.214*(I)+2954.43
1400.         CONTINUE
1410.         DO 35 I=17,M
1420.         U(1,I)=0.000175*(I)**0.00384
1430.         U(2,I)=7.9125*(I)+821.81
1440.         U(3,I)=27.063*(I)+5016.1
1450.         CONTINUE
1460.         U(1,N)=0.00529
1470.         U(2,N)=623.0
1480.         U(3,N)=562.9
1490.         DU(1,N)=0.
1500.         DU(2,N)=0.
1510.         DU(3,N)=0.
1520.         ...

```

```

1310***** NOW SET UP STREAM BOUNDARY CONDITIONS
1320* UC(1,1)=0.
1320* UC(2,1)=0.
1330* UC(3,1)=0.
1340* DO 120 I=1,N
1350*
1360* CP(1)=SQRT(GAMMAXUPX((3,I)/UPX(1,I)))
1370- CONTINUE
1380* CALL FILLSXUPX,CP,N,S,ALPH)
1390* CALL FILLDUXUPX,N,CB,SIG,D)
1400* CALL FILLD1D(SIG,D1)
1410* CALL URUFMXUPX,AR,Q8,UF,M)
1420*
1430* M*2
1440* CALL MATRX(SX,UC,Q1,N-1)
1450* CALL MATDX(D,Q1,Q2,N-1)
1460* CALL MATXS(S,Q2,Q3,R1)
1470- 130
1480* DT*CM*DX/(UPX(2,M)*CP(M))
1490* DLU(1,N)-DT*DXX(UF(1,M)-UF(1,M-1))
1500* DLU(2,M)-DT*DXX(UF(2,M)-UF(2,M-1))
1510* +DT*DXXUPX(3,M)*(AR(N)-AR(N-1))
1520* DLU(3,M)-DT*DXX(UF(3,M)-UF(3,M-1))
1530*
1540* DO 146 I=1,3
1541* Q4(1,M)-DLU(1,M)+DT*DXXQ3(I,M-1)
1550- 140
1560* CONTINUE
1570* CALL MATX(D11,05,M)
1580* CALL MATXS(06,UC,M)
1590* CALL MATDX(06,07,M)
1600* CALL MATXS(07,03,M)
1610* 2*M
1620* IF(2.EQ.N1) GO TO 150
1630- M+1
1640* GO TO 130
1650- 150
1660- CONTINUE
1670* DO 160 I=1,3
1680- U(R,I,J)=0.5*(UR(I,J)+UP(I,J)+UC(I,J))
1690- 160
1700- CONTINUE
1710- DO 171 I=1,N1
1720- Q8(1,I)*AR(1,I)*AR(1,I)
1730- Q8(2,I)*UR(2,I)/(AR(1)*QB(1,I))
1740- Q8(3,I)*UR(3,I)*0.4*AR(1)-Q8(1,I)*QB(2,I)
1750- 170
1760- CONTINUE
1770- TESTU-Q8(2,N1)-U(2,N1)
1780- TESTR-Q8(1,N1)-U(1,N1)
1790- TESTP-Q8(3,N1)-U(3,N1)
N-N
1800- DO 180 I=1,N1
1810- U(1,I)*UR(1,I)/AR(1,I)
1820- U(2,I)*UR(2,I)/(AR(1)*U(1,I))
1830- U(3,I)*UR(3,I)*0.4*AR(1)-U(1,I)*U(2,I)*U(3,I)
1840- 180
1850- CONTINUE
1860- U(1,I)-UPX(1,1)
1870- U(2,I)-UPX(2,1)
1880- U(3,I)-UPX(3,1)
1890- U(1,2)-UPX(1,2)
1900- U(2,2)-UPX(2,2)
1910- ***** NOW SET UP DOWN STREAM BOUNDARY CONDITIONS
1920- RHOC=UPX(1,N)*CP(N)
1930- U(1,N)-U(1,N1)+1/(CP(N)*2)*(5692.6-U(3,N1))

```

```

1940- UC(2,N)=U(2,N1)+1/RHOC*(U(3,N1)-5692.6)
1950- UC(3,N)=5692.6
1960- BC*AR(N)*X(U(3,N))/0.4+U(1,M)*X(U(2,N))X
1970- U(2,N)/2.)
1980- DU(1,N)*IC(1,N1)
1990- DU(2,N)*UC(2,N1)
2000- DU(3,N)*B-UP(3,N)
2010***** NOW GO BACK THROUGH THE LOOP AGAIN UNTIL MAX 3222
2020- ***** NOW GO BACK THROUGH THE LOOP AGAIN UNTIL MAX 3222
2030- ***** NOW CALCULATE THE FINAL MACH NUMBER
2040- IF(TESTU.NE.0) GO TO 999
2050- IF(TESTP.NE.0) GO TO 999
2060- IF(TESTR.EQ.0) GO TO 111
2070- 999
2080- CONTINUE
2090- 111
2100- CALL SECOND(CPF)
2110- CPT*CPF-CPI
2120- ***** NOW CALCULATE THE FINAL MACH NUMBER
2130- DO 206 I=1,N
2140- U8(I)=U2,I)/(SQRT(1.4*XU(3,I)/U1,I)))
2150- 200
2160- ***** NOW SET UP THE OUTPUT FORMAT
2170- CALL UINDO(XMIN,XMAX,YMIN,YMAX)
2180- CALL UAIN(CHAR)
2190- XX*X(1)
2200- YY*W(1)
2210- CALL UMDUE(XX,YY)
2220- DO 205 I=2,N
2230- XX-X(I)
2240- YY*W(1)
2250- CALL UMDU(XX,YY)
2260- CALL UEN(XX,YY)
2270- 205
2280- CALL UAIN(CHAR)
2290- YY-MAX-1.
2300- CALL UINDO(XMIN,XMAX,YMIN,YMAX)
2310- CALL UAIN(CHAR)
2320- XX-X(1)
2330- PSTAG=10000.
2340- YY-U(3,I)*PSTAG
2350- CALL UMDUE(XX,YY)
2360- DO 206 I=2,N
2370- XX-X(I)
2380- YY-U(3,I)*PSTAG
2390- CALL UEN(XX,YY)
2400- CONTINUE
2410- 206
2420- CALL UAIN(CHAR)
2430- WRITE(6,220)
2440- DO 210 I=1,N
2450- WRITE(6,230) X(I),U(2,I),U(3,I),U(1,I)
2460- 210
2470- FORMAT(/,4X,NUMBER OF ITERATIONS - 16,
2480- STOP
2490- 230
2500- ..

```

```

3140-      SUBROUTINE FILLD1(D1,SIG,D22)
3150-      DIMENSION D1(3,3,26),D22(3,3,26)
3160-      DO 440 I=1,26
3170-        D22(1,1,1)-1/1+SIG*D1(1,1,1)
3180-        D22(2,2,1)-1/1+SIG*D1(2,2,1)
3190-        D22(3,3,1)-1/1+SIG*D1(3,3,1)
3200-      END
3210-      D22(1,1,1)=0.
3220-      D22(2,1,1)=0.
3230-      D22(3,1,1)=0.
3240-      D22(1,2,1)=0.
3250-      D22(3,2,1)=0.
3260-      CONTINUE
3270-      RETURN
3280-      END
3290-      **** SUBROUTINE NUMBER FIVE ****
3300-      **** SUBROUTINE U(UA,DU,RS,N)
3310-      **** SUBROUTINE MATRIX(UA,DU,RS,N)
3320-      **** SUBROUTINE MAIX(SA,DUA,RS(3,26))
3330-      DIMENSION SA(3,3,26),DUA(3,26),RS(3,26)
3340-      DO 800 I=1,3
3350-        RS(I,I)=0.
3360-      CONTINUE
3370-      DO 820 J=1,3
3380-        RS(I,J)=SA(I,J,M)*DUA(J,M)+RS(I,M)
3390-      CONTINUE
3400-      RETURN
3410-      END
3420-      **** SUBROUTINE NUMBER SIX ****
3430-      **** SUBROUTINE U(UA,AE,U,UF,N)
3440-      **** SUBROUTINE URUF(UA,AE,U,UF,N)
3450-      **** SUBROUTINE U(U(3,N))
3460-      **** NOW SET UP THE U MATRIX
3470-      DIMENSION U(3,N)
3480-      DO 840 I=1,26
3490-        U(I,1)=E(I)*XUB(1,1)
3500-        U(I,2)=E(I)*XUB(1,2)
3510-        U(I,3)=E(I)*XUB(1,3)
3520-      CONTINUE
3530-      **** NOW SET UP THE F MATRIX
3540-      UF(1,1)=U(2,1)
3550-      UF(2,1)=0.4*XU(3,1)+XU(2,1)*XU(2,1)*0.8/XU(1,1)
3560-      UF(3,1)=1.4*XU(3,1)*XU(2,1)*XU(1,1)*0.2*XU(2,1)
3570-      CONTINUE
3580-      RETURN
3590-      END
3600-      **** SUBROUTINE NUMBER THREE ****
3610-      **** SUBROUTINE FILLD(U7,N,C1,SIG,D1)
3620-      DIMENSION C1(26),U7(3,26),D1(3,3,26)
3630-      DO 430 I=1,26
3640-        D1(1,1,1)=0.
3650-        D1(1,2,1)=0.
3660-        D1(1,3,1)=0.
3670-        D1(2,1,1)=0.
3680-        D1(2,2,1)=0.
3690-        D1(2,3,1)=0.
3700-        D1(3,1,1)=0.
3710-        D1(3,2,1)=0.
3720-        D1(3,3,1)=0.
3730-      CONTINUE
3740-      RETURN
3750-      END
3760-      **** SUBROUTINE NUMBER FOUR ****
3770-      **** SUBROUTINE FILLS(XU,C1,N,SX1)
3780-      DIMENSION U7(3,26),C1(26),SX1(3,3,26)
3790-      DO 420 I=1,26
3800-        SX1(1,1,1)-(1-U7(2,1))*U7(2,1)*0.2*(C1(I))C1(I))
3810-        SX1(1,2,1)-(U7(2,1)*0.4*(C1(I))C1(I)))
3820-        SX1(1,3,1)-(-0.4*(C1(I))C1(I)))
3830-        SX1(2,1,1)-(-C1(I))*U7(2,1)*U7(2,1)*0.2)
3840-        SX1(2,2,1)-C1(I)*U7(2,1)*0.4)
3850-        SX1(2,3,1)-0.4
3860-        SX1(3,1,1)-(C1(I)*U7(2,1)*U7(2,1)*0.2)
3870-        SX1(3,2,1)-(C1(I))*U7(2,1)*0.4)
3880-        SX1(3,3,1)-0.4
3890-      CONTINUE
3900-      RETURN
3910-      END
3920-      **** SUBROUTINE FILLD(U7,N,C1,SIG,D1)
3930-      DIMENSION C1(26),U7(3,26),D1(3,3,26)
3940-      DO 430 I=1,26
3950-        D1(1,1,1)=0.
3960-        D1(1,2,1)=0.
3970-        D1(1,3,1)=0.
3980-        D1(2,1,1)=0.
3990-        D1(2,2,1)=0.
4000-        D1(2,3,1)=0.
4010-        D1(3,1,1)=0.
4020-        D1(3,2,1)=0.
4030-        D1(3,3,1)=0.
4040-      CONTINUE
4050-      RETURN
4060-      END

```

VITA

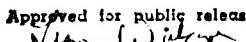
Charles Mark Allen was born February 7, 1956 in Salt Lake City, Utah. He attended Jackson Hole High School, Jackson, Wyoming, where he graduated in 1974. The years 1975-1977 were spent in Thailand as a missionary for the Church of Jesus Christ of Latter-Day Saints. Through this experience he learned to read, write and speak Thai fluently. Upon return to the United States he entered Brigham Young University, Provo, Utah, where he received a Bachelor of Science degree in Mechanical Engineering in December, 1980.

Lt. Allen received his commission in the USAF after finishing the OTS program in April, 1981. He then entered the School of Engineering, Air Force Institute of Technology in June 1981. Upon completion of the AFIT Aeronautical Masters curriculum Lt. Allen has been assigned to the Simulator Validation Branch of the Foreign Technology Division, Wright-Patterson AFB, Ohio.

Permanent address: 762 Washington Street
Montpelier, Idaho 83254

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

| REPORT DOCUMENTATION PAGE | | READ INSTRUCTIONS BEFORE COMPLETING FORM |
|---|--------------------------------|---|
| 1. REPORT NUMBER AFIT/GAS/82D - 2 AEI | 2. GOVT ACCESSION NO. | 3. RECIPIENT'S CATALOG NUMBER |
| 4. TITLE (and Subtitle) IMPLICIT NUMERICAL SOLUTION FOR A NORMAL SHOCK | | 5. TYPE OF REPORT & PERIOD COVERED MS Thesis |
| | | 6. PERFORMING ORG. REPORT NUMBER |
| 7. AUTHOR(s) Charles Mark Allen 1st Lt. USAF | 8. CONTRACT OR GRANT NUMBER(s) | |
| 9. PERFORMING ORGANIZATION NAME AND ADDRESS Air Force Institute of Technology (AFIT/EN) Wright-Patterson AFB, Ohio 45433 | | 10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS |
| 11. CONTROLLING OFFICE NAME AND ADDRESS | | 12. REPORT DATE May 1983 |
| | | 13. NUMBER OF PAGES 80 |
| 14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) | | 15. SECURITY CLASS. (of this report) Unclassified |
| | | 15a. DECLASSIFICATION DOWNGRADING SCHEDULE |
| 16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited | | |
| 17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report) | | |
| 18. SUPPLEMENTARY NOTES <p style="text-align: right;">Approved for public release: IAW AFR 190-1A  LYNNE W. CLAVER Dean for Research and Professional Development Air Force Institute of Technology (AFTC) Wright-Patterson AFB OH 45433 9 SEP 1983</p> | | |
| 19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Numerical Solution of Navier-Stokes Equations Implicit Numerical Techniques Characteristic Boundary Conditions | | |
| 20. ABSTRACT (Continue on reverse side if necessary and identify by block number) <p style="text-align: center;">There are basically two techniques used to solve the Navier-Stokes equations for fluid flow. These techniques are implicit and explicit methods. Both methods along with characteristic boundary conditions for solving quasi-one-dimensional nozzle flow are presented. Two types of characteristic boundary conditions, those of Steger and McKenna, were tested for each scheme. Solutions of isentropic supersonic flow and flow with shocks were obtained for a diverging nozzle. The behavior of each boundary condition on both implicit and explicit schemes were the same. They deviated from the</p> | | |

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

theoretical values by less than one percent. A test to determine the utility of each scheme was run by allowing the exit boundary conditions to change in the hope of forcing the shock to move upstream or downstream. The shock would not move in the implicit scheme for either boundary condition. The explicit scheme could move the shock, but only when Steger's boundary condition was used. (This is the one which specified only pressure.)

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

